Proceedings of the
12th Japanese-Hungarian Symposium
on Discrete Mathematics and Its Applications
March 21-24, 2023, Budapest, Hungary

Editors:

Tibor Jordán
Department of Operations Research,
ELTE Eötvös Loránd University
and
ELKH-ELTE Egerváry Research Group
Eötvös Loránd Research Network (ELKH)
tibor.jordan@ttk.elte.hu

Gyula Y. Katona
Department of Computer Science and Information Theory
Budapest University of Technology and Economics
and
ELKH-ELTE Numerical Analysis and Large Networks Research Group
Eötvös Loránd Research Network (ELKH)
katona.gyula@vik.bme.hu

Csaba Király
ELKH-ELTE Egerváry Research Group
Eötvös Loránd Research Network (ELKH)
csaba.kiraly@ttk.elte.hu

Gábor Wiener
Department of Computer Science and Information Theory
Budapest University of Technology and Economics
wiener@cs.bme.hu

ISBN 978-963-421-903-3

Cover design: Kazuhiko Shiozaki, 1999

Contents

Preface

The present volume consists of the papers and extended abstracts of the talks presented at the 12th Japanese-Hungarian Symposium on Discrete Mathematics and Its Applications (Budapest, March 21-24, 2023). Based on a long history of cooperation among Japanese and Hungarian scientists in the area of discrete mathematics, the first joint symposium was announced 25 years ago to be held in Kyoto (March 17-19, 1999). The subsequent symposia in the series took place in Budapest (April 20-23, 2001), Tokyo (January 21-24, 2003), Budapest (June 3-6, 2005), Sendai (April 3-5, 2007), Budapest (May 16-19, 2009), Kyoto (May 31 - June 3, 2011), Veszprém (June 4-7, 2013), Fukuoka (June 2-5, 2015), Budapest (May 22-25, 2017), and Tokyo (May 27-30, 2019).
The 12th Symposium was jointly organized by the Department of Operations Research, ELTE Eötvös Loránd University, Budapest and by the Department of Computer Science and Information Theory, Budapest University of Technology and Economics.

Advisory Board

András Frank (Department of Operations Research, ELTE Eötvös Loránd University)
Satoru Fujishige (Research Institute for Mathematical Sciences, Kyoto University)
Satoru Iwata (Department of Mathematical Informatics, University of Tokyo)
Tibor Jordán (Department of Operations Research, ELTE Eötvös Loránd University)
Naoki Katoh (Graduate School of Information Science, University of Hyogo)
Gyula Y. Katona (Department of Computer Science and Information Theory, Budapest University of Technology and Economics)
Tamás Király (Department of Operations Research, ELTE Eötvös Loránd University)
Kazuo Murota (The Institute of Statistical Mathematics, and Faculty of Economics and Business Administration, Tokyo Metropolitan University)
András Recski (Department of Computer Science and Information Theory, Budapest University of Technology and Economics)
Takeshi Tokuyama (Department of Computer Science, Kwansei Gakuen University)

Invited speakers

Kristóf Bérczi (Department of Operations Research, ELTE Eötvös Loránd University)
Tamás Fleiner (Department of Computer Science and Information Theory, Budapest University of Technology and Economics)
Hiroshi Hirai (Department of Mathematical Informatics, The University of Tokyo)
Naonori Kakimura (Department of Mathematics, Keio University)
Kazuhisa Makino (Research Institute for Mathematical Sciences, Kyoto University)
Shin-ichi Tanigawa (Department of Mathematical Informatics, University of Tokyo)
Takeshi Tokuyama (Department of Computer Science, Kwansei Gakuin University)
Yutaro Yamaguchi (Department of Information and Physical Sciences, Osaka University)

Organizing Committee

Kristóf Bérczi (Department of Operations Research, ELTE Eötvös Loránd University)
Tibor Jordán (Department of Operations Research, ELTE Eötvös Loránd University)
Gyula Y. Katona (Department of Computer Science and Information Theory, Budapest University of Technology and Economics)
Csaba Király (ELKH-ELTE Egerváry Research Group, Eötvös Loránd Research Network (ELKH))
Gábor Wiener (Department of Computer Science and Information Theory, Budapest University of Technology and Economics)

The conference was supported by the National Research, Development and Innovation Fund of the Ministry of Innovation and Technology, the Aquincum Institute of Technology (AIT), the Doctoral School of Mathematics, ELTE Eötvös Loránd University, and the Faculty of Electrical Engineering and Informatics, Budapest University of Technology and Economics.

The organizers wish to thank all the contributors for submitting papers, and all their colleagues, graduate students and sponsors for their assistance and support.

Budapest, March 7, 2023.

Dynamic pricing schemes

Kristóf Bérczi[1]

MTA-ELTE Matroid Optimization
Research Group
ELKH-ELTE Egerváry Research Group
Department of Operations Research
Eötvös Loránt University
Budapest, Hungary
kristof.berczi@ttk.elte.hu

Abstract:

A combinatorial market consists of a set of indivisible items and a set of agents, where each agent has a valuation function that specifies for each subset of items its value for the given agent. From an optimization point of view, the goal is usually to determine a pair of pricing and allocation of the items that provides an efficient distribution of the resources, i.e., maximizes the social welfare, or is as profitable as possible for the seller, i.e., maximizes the revenue. Dynamic pricing schemes were introduced as an alternative to posted-price mechanisms. In contrast to static models, the dynamic setting allows to update the prices between agent-arrivals based on the remaining sets of items and agents, and so it is capable of maximizing social welfare without the need for a central coordinator.

In this talk, we overview recent results on the existence of optimal dynamic prices, with particular emphasis on the case of matroid rank valuations. For the case of two agents with matroid rank valuations, we give polynomial-time algorithms that always find such prices when one of the matroids is a partition matroid or both matroids are strongly base orderable, thus partially answering a question raised by Dütting and Végh. For multi-demand valuations, we propose an approach that is based on computing an optimal dual solution of the maximum social welfare problem with distinguished structural properties. By relying on an optimal dual solution, we show the existence of optimal dynamic prices in unit-demand markets, multi-demand markets up to three agents, and bi-demand valuations with an arbitrary number of agents. Finally, we study the existence of optimal dynamic prices under fairness constraints in unit-demand markets. We propose four possible notions of envy-freeness depending on the time period over which agents compare themselves to others: the entire time horizon, only the past, only the future, or only the present.

Keywords: Algorithms, Dynamic pricing scheme, Envy-free allocations, Revenue maximization, Social welfare maximization

1 Introduction

A combinatorial market consists of a set of indivisible goods and a set of agents, where each agent has a valuation function that represents the agent's preferences over the subsets of items. From an optimization point of view, the goal is to find an allocation of the items to agents in such a way that the total sum

[1]The talk is based on joint works with Naonori Kakimura and Yusuke Kobayashi [3], Erika Bérczi-Kovács and Evelin Szögi [1], and Laura Codazzi, Julian Golak and Alexander Grigoriev [2]. The work was supported by the Lendület Programme of the Hungarian Academy of Sciences – grant number LP2021-1/2021 and by the Hungarian National Research, Development and Innovation Office – NKFIH, grant numbers FK128673 and TKP2020-NKA-06.

of the agents' values is maximized – this sum is called the social welfare. An optimal allocation can be found efficiently in various settings [5, 8, 12, 14], but the problem becomes significantly more difficult if one would like to realize the optimal social welfare through simple mechanisms.

A great amount of work concentrated on finding optimal pricing schemes. Given a price for each item, we define the utility of an agent for a bundle of items to be the value of the bundle with respect to the agent's valuation, minus the total price of the items in the bundle. A pair of pricing and allocation is called a Walrasian equilibrium if the market clears (that is, all the items are assigned to agents) and everyone receives a bundle that maximizes her utility. Given any Walrasian equilibrium, the corresponding price vector is referred to as Walrasian pricing, and the definition implies that the corresponding allocation maximizes social welfare.

Although Walrasian equilibria have distinguished properties, Cohen-Addad et al. [6] observed that Walrasian prices are not powerful enough to control the market on their own. The reason is that ties among different bundles must be broken in a coordinated fashion that is consistent with maximizing social welfare. Furthermore, this problem cannot be resolved by finding Walrasian prices where ties do not occur as [10] showed that minimal Walrasian prices necessarily induce ties.

Dynamic pricing schemes were introduced as an alternative to posted-price mechanisms that are capable of maximizing social welfare even without a central tie-breaking coordinator. In this model, the agents arrive in a sequential order, and each agent selects a bundle of the remaining items that maximizes her utility. The agents' preferences are known in advance, and the seller is allowed to update the prices between agent arrivals based upon the remaining set of items, but without knowing the identity of the next agent. The main open problem in [6] asked whether any market with gross substitutes valuations has a dynamic pricing scheme that achieves optimal social welfare.

2 Preliminaries

A combinatorial market consists of a set S of *indivisible items* and a set A of *agents*. Throughout the talk, we denote by $m := |S|$ and $n := |A|$ the numbers of items and agents, respectively. An *allocation* $\mathbf{X}$ assigns each agent a a subset X_a of items so that each item is assigned to at most one agent.

In a *unit-demand market*, each agent $a \in A$ has a valuation $v_a \colon S \to \mathbb{R}_+$ over individual items and she desires only a single good, that is, we consider allocations $\mathbf{X}$ with $|X_a| \leq 1$ for $a \in A$ – in such cases we denote the item obtained by agent a by x_a. We always assume that the agents' valuations are known in advance. Furthermore, we assume that $v_a(\emptyset) = 0$ for all agents $a \in A$. Given prices $p(s)$ for each item $s \in S$, the *utility* of agent a for item s is $u_a(s) := v_a(s) - p(s)$. Then the *social welfare* corresponding to the allocation is $\sum_{a \in A} v_a(x_a)$, while the *revenue* of the seller is $\sum_{a \in A} p(x_a)$.

In a *static pricing scheme*, the seller sets the price $p(s)$ of each item $s \in S$ in advance. Two fundamental problems in combinatorial markets are to find a pair of pricing vector $p \colon S \to \mathbb{R}_+$ and allocation $\mathbf{X}$ such that the social welfare or the revenue is maximized. In contrast, in a *dynamic pricing scheme* the agents arrive one after the other, and the seller can update the prices between their arrivals based on the remaining sets of items and agents. The order in which agents arrive is represented by a bijection $\sigma \colon A \to [n]$. The sets of agents, items and prices available before the arrival of the tth agent are denoted by A_t, S_t and p_t, respectively. The utility of agent a for item s at time step t is then defined as $u_{a,t}(s) := v_a(s) - p_t(s)$. The next agent always chooses an item that maximizes her utility. After the last agent has left, the pricing scheme terminates and results in pricing vectors $\mathbf{p} = (p_1, \ldots, p_n)$ and an allocation $\mathbf{X} = (x_1, \ldots, x_n)$, where p_t is the price vector available at the arrival of the tth agent and x_t is the item allocated to her. Note that x_t might be an empty set if the utility of the agent is non-positive for each item in S_t. We call a dynamic pricing scheme *optimal* if the final allocation maximizes the objective, that is, the social welfare or the revenue, irrespective of the order in which the agents arrived.

3 Matroid rank valuations

The result of this section appeared in [3]. As a starting step towards understanding the general case, we consider the existence of a static pricing scheme for a two-agent market with matroid rank valuations, because a matroid rank function is a fundamental example of gross substitutes valuations. Here, a matroid with a ground set S and a base family $\mathcal{B}$ is denoted by $M = (S, \mathcal{B})$ and we denote $p(X) := \sum_{s \in X} p(s)$ for $p : S \to \mathbb{R}$ and $X \subseteq S$. In particular, we concentrate on the following conjecture.

Conjecture 1 *Let $M_1 = (S, \mathcal{B}_1)$ and $M_2 = (S, \mathcal{B}_2)$ be matroids with rank functions r_1 and r_2, respectively. Then, there exists a function $p : S \to \mathbb{R}$ (called a* price vector*) satisfying the following conditions.*

1. *For $B_1 \in \arg\max_{X \subseteq S}(r_1(X) - p(X))$ and $B_2 \in \arg\max_{Y \subseteq S \setminus B_1}(r_2(Y) - p(Y))$, we have $r_1(B_1) + r_2(B_2) = \max\{r_1(X) + r_2(Y) \mid X, Y \subseteq S,\ X \cap Y = \emptyset\}$.*
2. *For $B_2 \in \arg\max_{Y \subseteq S}(r_2(Y) - p(Y))$ and $B_1 \in \arg\max_{X \subseteq S \setminus B_2}(r_1(X) - p(X))$, we have $r_1(B_1) + r_2(B_2) = \max\{r_1(X) + r_2(Y) \mid X, Y \subseteq S,\ X \cap Y = \emptyset\}$.*

This conjecture can be interpreted as follows. There are two agents and each agent $i \in \{1, 2\}$ has a matroid rank valuation function r_i. If agent i comes to a shop first, then she chooses an arbitrary bundle B_i that maximizes her utility $r_i - p$, and the second agent chooses a best bundle in $S \setminus B_i$. The requirements mean that any choice of B_i results in an allocation maximizing the social welfare. Thus, whoever comes first, we can achieve the optimal social welfare.

It turns out that Conjecture 1 can be reduced to the following.

Conjecture 2 *Let $M_1 = (S, \mathcal{B}_1)$ and $M_2 = (S, \mathcal{B}_2)$ be matroids with a common ground set S such that there exist disjoint bases $B_1 \in \mathcal{B}_1$ and $B_2 \in \mathcal{B}_2$ with $B_1 \cup B_2 = S$. Then, there exists a function $p : S \to \mathbb{R}$ (called a* price vector*) satisfying the following conditions.*

1. *For $B_1 \in \arg\min_{X \in \mathcal{B}_1} p(X)$, we have $S \setminus B_1 \in \mathcal{B}_2$.*
2. *For $B_2 \in \arg\min_{X \in \mathcal{B}_2} p(X)$, we have $S \setminus B_2 \in \mathcal{B}_1$.*

In the conjecture, there are two agents and each agent $i \in \{1, 2\}$ wants to buy a set of items that forms a basis in $\mathcal{B}_i$. If agent i comes to a shop first, then she chooses a cheapest set B_i in $\mathcal{B}_i$ with an arbitrary tie-breaking rule. The requirements mean that, regardless of the choice of B_i, the remaining set $S \setminus B_i$ is a desired set for the other agent.

Conjecture 2 was first suggested by Dütting and Végh [7] in a form where the price vector p is ought to have all different values, that is, $p(s_1) \neq p(s_2)$ for $s_1 \neq s_2$, which implies that $B_i \in \arg\min_{X \in \mathcal{B}_i} p(X)$ is unique for $i = 1, 2$. However, this difference is not essential, because we can apply a perturbation to p without affecting the requirements in Conjecture 2.

While Conjecture 2 remains open in general, we give polynomial-time algorithms for two important special cases: when one of the matroids is a partition matorid, and when both matroids are strongly base orderable.

Theorem 3 *If M_1 is a partition matroid and M_2 is an arbitrary matroid, then Conjectures 1 and 2 hold, and a price vector p satisfying the conditions can be computed in polynomial time.*

Theorem 4 *If both M_1 and M_2 are strongly base orderable, then Conjectures 1 and 2 hold. Furthermore, a price vector p satisfying the conditions can be computed in polynomial time if, for any pair of bases, the bijection between them can be computed in polynomial time.*

We further show the equivalence between Conjecture 2 and its weighted counterpart as below.

Conjecture 5 *For $i \in \{1, 2\}$, let $M_i = (S, \mathcal{B}_i)$ be a matroid and $w_i : S \to \mathbb{R}$ be a weight function. Assume that there exist disjoint bases $B_1 \in \mathcal{B}_1$ and $B_2 \in \mathcal{B}_2$ with $B_1 \cup B_2 = S$. Then, there exists a function $p : S \to \mathbb{R}$ satisfying the following conditions.*

1. *For* $B_1 \in \arg\max_{X \in \mathcal{B}_1}(w_1(X) - p(X))$*, we have that* B_1 *is a maximizer of* $w_1(X) + w_2(S \setminus X)$ *subject to* $X \in \mathcal{B}_1$ *and* $S \setminus X \in \mathcal{B}_2$*.*

2. *For* $B_2 \in \arg\max_{X \in \mathcal{B}_2}(w_2(X) - p(X))$*, we have that* B_2 *is a maximizer of* $w_1(S \setminus X) + w_2(X)$ *subject to* $S \setminus X \in \mathcal{B}_1$ *and* $X \in \mathcal{B}_2$*.*

Clearly, Conjecture 2 is a special case of Conjecture 5; this follows easily by setting $w_1 \equiv w_2 \equiv 0$. Somewhat surprisingly, the reverse implication also holds for arbitrary matroids.

Theorem 6 *If Conjecture 2 is true, then Conjecture 5 is also true.*

Based on Theorem 6 and the properties of partition and strongly base orderable matroids, we have the following corollaries.

Corollary 7 *If* M_1 *is a partition matroid and* M_2 *is an arbitrary matroid, then Conjecture 5 holds, and a price vector* p *satisfying the conditions can be computed in polynomial time.*

Corollary 8 *If both* M_1 *and* M_2 *are strongly base orderable, then Conjecture 5 holds. Furthermore, a price vector* p *satisfying the conditions can be computed in polynomial time if, for any pair of bases, the bijection between them can be computed in polynomial time.*

Finally, we prove that Theorem 6 can be generalized to gross substitutes valuations, i.e., $\mathrm{M}^\natural$-concave functions.

4 Unit- and bi-demand markets

The result of this section appeared in [1]. In multi-demand markets, each agent t has a positive integer bound $b(t)$ on the number of desired items, and the value of a set is the sum of the values of the $b(t)$ most valued items in the set. In particular, if we set each $b(t)$ to one or two then we get the unit-demand or bi-demand cases, respectively.

For multi-demand markets, the problem of finding an allocation that maximizes social welfare is equivalent to a maximum weight b-matching problem in a bipartite graph with vertex classes corresponding to the agents and items, respectively. The high level idea of our approach is to consider the dual of this problem, and to define an appropriate price vector based on an optimal dual solution with distinguished structural properties.

Based on the primal-dual interpretation of the problem, first we give a simpler proof of a result of Cohen-Addad et al. [6] on unit-demand valuations.

Theorem 9 (Cohen-Addad et al.) *Every unit-demand market admits an optimal dynamic pricing that can be computed in polynomial time.*

When the total demand of the agents exceeds the number of available items, ensuring the optimality of the final allocation becomes more intricate. Therefore, we consider instances satisfying the following property:

each agent $t \in T$ receives exactly $b(t)$ items in every optimal allocation. (OPT)

While this is a restrictive assumption, it is a reasonable condition that holds for a wide range of applications, and also appeared in [4] and recently in [13]. For example, if the total number of items is not less than the total demand of the agents and the value of each item is strictly positive for each agent, then it is not difficult to check that (OPT) is satisfied.

The problem becomes significantly more difficult for larger demands. Berger et al. [4] observed that bundles that are given to an agent in different optimal allocations satisfy strong structural properties. For markets up to three multi-demand agents, they grouped the items into at most eight equivalence classes based on which agent could get them in an optimal solution, and then analyzed the item-equivalence

graph for obtaining an optimal dynamic pricing. We show that, when assumption (OPT) is satisfied, these properties follow from the primal-dual interpretation of the problem, and give a new proof of their result for such instances.

Theorem 10 (Berger et al.) *Every multi-demand market with property* (OPT) *and at most three agents admits an optimal dynamic pricing scheme, and such prices can be computed in polynomial time.*

Finally, we give an algorithm for determining optimal dynamic prices in bi-demand markets with an arbitrary number of agents, that is, when the demand $b(t)$ is two for each agent t.[1] Besides structural observations on the dual solution, the proof relies on uncrossing sets that are problematic in terms of resolving ties.

Theorem 11 *Every bi-demand market with property* (OPT) *admits an optimal dynamic pricing scheme, and such prices can be computed in polynomial time.*

5 Dynamic pricing under fairness constraints

The result of this section appeared in [2]. The original motivation behind dynamic pricing schemes was to shift the tie-breaking process from the central coordinator to the customers, as in reality customers choose bundles of items without caring about social optimum. As we have seen in the previous sections, the dynamic setting is indeed capable of maximizing social welfare without the need for a central coordinator. On the other hand, this approach has an implication on the fairness of the final allocation that is usually not emphasized. The model assumes that the customers' sole objective is to pick a bundle of items maximizing their utility with respect to the prices available at their arrival, and they are not concerned with prices at earlier and/or later times. This means that envy-freeness is ensured only locally, and the final allocation together with the prices at which the items were bought do not necessarily form an envy-free solution over all time horizon.

The model we consider here differs from earlier ones mainly in that we are seeking for optimal pricing schemes under fairness constraints. In the static setting, a pair of pricing p and allocation $\mathbf{x}$ is *envy-free* if $x_a \in \arg\max\{u_a(s) \mid s \in S\}$ holds for each agent $a \in A$. The dynamic setting naturally suggests variants in which envy-freeness is defined over a subset of time steps. Let $T_a \subseteq [n]$ be a subset of time steps for each agent $a \in A$. Then price vectors $\mathbf{p} = (p_1, \dots, p_n)$ and allocation $\mathbf{X} = (x_1, \dots, x_n)$ form an envy-free allocation if $x_a \in \arg\max\{u_{a,t}(s) \mid t \in T_a, s \in S_t\}$ for each agent $a \in A$. We propose four possible notions of envy-freeness of different strength depending on the time period over which agents compare themselves to others:

(F1) *Strong envy-freeness.* Agents consider prices for the whole time horizon, that is, $T_a = \{1, \dots, n\}$ for $a \in A$.

(F2) *Ex-post envy-freeness.* Agents consider prices available after and at their arrival, that is, $T_a = \{\sigma(a), \dots, n\}$ for $a \in A$.

(F3) *Ex-ante envy-freeness.* Agents consider prices available before and at their arrival, that is, $T_a = \{1, \dots, \sigma(a)\}$ for $a \in A$.

(F4) *Weak envy-freeness.* Agents consider prices at their arrival, that is, $T_a = \{\sigma(a)\}$ for $a \in A$.

Using this terminology, optimal dynamic pricing schemes discussed in [1, 3, 4, 6, 13] provide weakly envy-free solutions. It is worth mentioning that, though at first sight they might seem to be symmetric, the ex-post and ex-ante cases turns out to behave quite differently.

We distinguish further variants of the model depending on whether ties between items are broken by the seller or the agents:

[1] Recently, Pashkovich and Xie [13] showed that the result of Berger et al. [4] can be generalized from three to four agents. They further extended the results of the current paper on bi-demand valuations to the case when each agent is ready to buy up to three items.

(C1) *Seller-chooses.* If there are several items maximizing the utility of the current agent, then the seller decides which one to allocate to her.

(C2) *Agent-chooses.* If there are several items maximizing the utility of the current agent, then she decides which one to take.

In terms of finding an optimal pricing, problem (C1) is easier. Indeed, given an optimal pricing for (C2), the seller can always decide to allocate the item that was chosen by the agent.

Previous works generally assumed that agents arrive in an unspecified order. Besides this, we consider two further variants based on the control and information of the arrival process:

(O1) *Unspecified.* The agents arrive in a fixed order that the seller has no information on.

(O2) *Predetermined.* The agents arrive in a fixed order that the seller knows in advance.

(O3) *Alterable.* The order of the agents is determined by the seller.

As for the *objective function*, we either consider the *social welfare* $W(\mathbf{X}) = \sum_{a \in A} v_a(x_a)$ or the *revenue* of all sold items $R(\mathbf{p}, \mathbf{X}) = \sum_{a \in A} p_{\sigma(a)}(x_a)$.

These variants and our results are summarized in Table 1. The results are split horizontally by the type of envy-freeness considered, while the columns are indexed by the type of the ordering of the agents. Algorithmic results hold irrespective of how agents break ties, while hardness results hold even if ties are broken by the seller. It is worth noting that the $O(\log(n))$-approximation algorithm of Guruswami et al. [9] extends to all of variants of envy-free pricing where the objective is to maximize the revenue.

Table 1: Complexity landscape of social welfare and revenue maximization under fairness constraints in unit-demand markets. Algorithmic results (green cells) hold even in the agent-chooses setting, while hardness results (red cells) hold already for the seller-chooses case. In each row, complexities of cells with light shade are implied by cells with darker shade.

		Welfare maximization			Revenue maximization		
	Ties	Unspecified	Predetermined	Alterable	Unspecified	Predetermined	Alterable
Strong	Agents	Not exists	Not exists	Not exists	Not exists	Not exists	Not exists
	Seller	P [11, 15]	P	P	APX-hard	APX-hard	APX-hard
Ex-post		P	P	P	APX-hard	APX-hard	P
Ex-ante		P	P	P	APX-hard	APX-hard	P
Weak		P [6, Thm. 3.1]	P	P	Open	P	P

References

[1] K. Bérczi, E. R. Bérczi-Kovács, and E. Szögi. A dual approach for dynamic pricing in multi-demand markets. *arXiv preprint arXiv:2107.05131*, 2021.

[2] K. Bérczi, L. Codazzi, J. Golak, and A. Grigoriev. Envy-free dynamic pricing schemes. *arXiv preprint arXiv:2301.01529*, 2023.

[3] K. Bérczi, N. Kakimura, and Y. Kobayashi. Market pricing for matroid rank valuations. *SIAM Journal on Discrete Mathematics*, 35(4):2662–2678, 2021.

[4] B. Berger, A. Eden, and M. Feldman. On the power and limits of dynamic pricing in combinatorial markets. In *International Conference on Web and Internet Economics*, pages 206–219. Springer, 2020.

[5] E. H. Clarke. Multipart pricing of public goods. *Public choice*, pages 17–33, 1971.

[6] V. Cohen-Addad, A. Eden, M. Feldman, and A. Fiat. The invisible hand of dynamic market pricing. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, pages 383–400, 2016.

[7] P. Dütting and L. A. Végh. Private Communication, 2017.

[8] T. Groves. Incentives in teams. *Econometrica: Journal of the Econometric Society*, pages 617–631, 1973.

[9] V. Guruswami, J. D. Hartline, A. R. Karlin, D. Kempe, C. Kenyon, and F. McSherry. On profit-maximizing envy-free pricing. In *SODA*, volume 5, pages 1164–1173, 2005.

[10] J. Hsu, J. Morgenstern, R. Rogers, A. Roth, and R. Vohra. Do prices coordinate markets? In *Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing*, pages 440–453, 2016.

[11] A. S. Kelso Jr and V. P. Crawford. Job matching, coalition formation, and gross substitutes. *Econometrica: Journal of the Econometric Society*, pages 1483–1504, 1982.

[12] N. Nisan and I. Segal. The communication requirements of efficient allocations and supporting prices. *Journal of Economic Theory*, 129(1):192–224, 2006.

[13] K. Pashkovich and X. Xie. A two-step approach to optimal dynamic pricing in multi-demand combinatorial markets. *arXiv preprint arXiv:2201.12869*, 2022.

[14] W. Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of Finance*, 16(1):8–37, 1961.

[15] L. Walras and E. d. P. Pure. Lausanne: L. *Corbaz & Cie*, 1874.

Division of goods and bads to many players

Tamás Fleiner[1]

Department of Computer Science and
Information Theory
Budapest University of
Technology and Economics
Magyar Tudósok körútja 2,
H-1117 Budapest, Hungary and
Institute of Economics,
Centre for Economic and Regional Studies,
Tóth Kálmán u. 4, H-1097 Budapest, Hungary
fleiner.tamas@vik.bme.hu

Abstract: The goal of this work is to recall some well-known facts about the fair division problem and cake-cutting protocols, to exhibit certain not so well-known results on proportional division with unequal shares and to formulate some open research problems.

Fair division, unequal shares, Robertson-Webb model, division protocol

1 Introduction

In cake cutting problems, the cake symbolizes a heterogeneous and divisible resource that shall be distributed among n players. Each player has her own measure function, which determines the value of any part of the cake for her. The aim of proportional cake cutting is to allocate each player a piece that is worth at least as much as her proportional share, evaluated with her measure function [26]. The measure functions are not known to the protocol. Instead of a cake, we may need to distribute chores. The difference from the cake is that it represents negative utility for the players, hence a player is better off if she gets less of it. Consequently, in a proportional chore division no player gets more of the chore than her proportional share.

The efficiency of a fair division protocol can be measured by the number of queries. In the standard Robertson-Webb model [22], two kinds of queries are allowed. The first one is the *cut* query, in which a player is asked to mark the cake at a distance from a given starting point so that the piece between these two is worth a given value to her. The second one is the *eval* query, in which a player is asked to evaluate a given piece according to her measure function.

If shares are meant to be *equal* for all players, then the proportional share is defined as $\frac{1}{n}$ of the whole cake. In the *unequal shares* version of the problem (also called cake cutting with entitlements), proportional share is defined as a player-specific demand, summing up to the value of the cake over all players.

The aim of this paper is to determine the query complexity of proportional cake cutting and chore division in the case of unequal shares. Robertson and Webb write in their seminal book [22] "Nothing approaching general theory of optimal number of cuts for unequal shares division has been given to date.

[1]Research was supported by OTKA grant K128611. The research reported in this work and carried out at the Budapest University of Technology and Economics was supported by the "TKP2020, National Challenges Program" of the National Research Development and Innovation Office (BME NC TKP2020 and OTKA124171) and by the Higher Education Excellence Program of the Ministry of Human Capacities in the frame of the Artificial Intelligence research area of the Budapest University of Technology and Economics (BME FIKP-MI/SC).

This problem may prove to be very difficult." Instead of the number of physical cuts, we now settle the issue for the number of queries, which is the standard measure of efficiency for cake cutting protocols.

We provide formal definitions in Section 2. In Section 3 we focus on our protocol for the problem [10]. The idea is that we recursively render the players in two batches so that these batches can simulate two players who aim to cut the cake into two approximately equal halves. Our protocol requires only $2\,(n-1)\cdot\lceil\log_2 D\rceil$ queries. Other known protocols reach $D\cdot\lceil\log_2 D\rceil$ and $n(n-1)\cdot\lceil\log_2 D\rceil$ queries, thus ours is the fastest procedure that derives a proportional division for the n-player cake cutting problem with unequal shares.

We complement our positive result by showing a lower bound of $\Omega\,(n\cdot\log D)$ on the query complexity of the proportional cake cutting problems in Section 4. Our proof generalizes, but does not rely on the lower bound proof given by Edmonds and Pruhs in [12] for the problem of proportional division with equal shares. In Section 5, we turn to irrational demands and present our protocol for the proportional cake cutting problem by reducing it to the same problem with integer demands only. We conclude in Section 6 with two open research problems.

2 Preliminaries

We begin with formally defining our input. Our setting includes a set of players of cardinality n, denoted by $\{P_1, P_2, \ldots, P_n\}$, and a heterogeneous and divisible good, which we refer to as the cake and project to the unit interval $[0,1]$. Each player P_i has a non-negative, absolutely continuous *measure function* μ_i that is defined on Lebesgue-measurable sets. We remark that absolute continuity implies that every zero-measure set has value 0 according to μ_i as well. In particular, $\mu_i((a,b)) = \mu_i([a,b])$ for any interval $[a,b] \subseteq [0,1]$. Besides measure functions, each player P_i has a *demand* $d_i \in \mathbb{Z}^+$, representing that P_i is entitled to receive $d_i / \sum_{j=1}^{n} d_j \in]0,1[$ share of the whole cake. The value of the whole cake is identical for all players, in particular it is the sum of all demands:

$$\forall 1 \le i \le n \quad \mu_i([0,1]) = D = \sum_{j=1}^{n} d_j.$$

We remark that an equivalent formulation is also used sometimes, where the demands are rational numbers that sum up to 1, the value of the full cake. Such an input can be transformed into the above form simply by multiplying all demands by the least common denominator of all demands. As opposed to this, if demands are allowed to be irrational numbers, then no ratio-preserving transformation might be able to transform them to integers. That is why the case of irrational demands is treated separately.

The cake $[0,1]$ will be partitioned into subintervals in the form $[x,y), 0 \le x \le y \le 1$. A finite union of such subintervals forms a *piece* X_i allocated to player P_i. We would like to stress that a piece is not necessarily connected.

Definition 1 *A set $\{X_i\}_{1\le i\le n}$ of pieces is a* division *of the cake $[0,1]$ if $\bigcup_{1\le i\le n} X_i = [0,1]$ and $X_i \cap X_j = \emptyset$ for all $i \neq j$. We call division $\{X_i\}_{1\le i\le n}$* proportional *if $\mu_i(X_i) \ge d_i$ for all $1 \le i \le n$. If we work with chore then proportionality of a division means that $\mu_i(X_i) \le d_i$ holds for all $1 \le i \le n$.*

In words, proportionality means that each player receives a piece with which her demand is satisfied. We do not consider Pareto optimality or alternative fairness notions such as envy-freeness in this paper.

We now turn to defining the measure of efficiency of a cake cutting protocol. We assume that $1 \le i \le n$, $x, y \in [0,1]$, and $0 \le \alpha \le 1$. Oddly enough, the Robertson-Webb query model was not formalized explicitly by Robertson and Webb first, but by Woeginger and Sgall [28], who attribute it to the earlier two. In their query model, a protocol can ask agents the following two types of queries.

- *Cut query* (P_i, α) returns the leftmost point x so that $\mu_i([0,x]) = \alpha$. In this operation x becomes a so-called *cut point*.

- *Eval query* (P_i, x) returns $\mu_i([0, x])$. Here x must be a cut point.

Notice that this definition implies that choosing sides, sorting marks or calculating any other parameter than the value of a piece are not counted as queries and thus they do not influence the efficiency of a protocol. Our protocols do not abuse the model by performing a large number of such operations. We also remark that the "leftmost point" criterion in the cut query can be omitted if the measure μ_i is not only absolutely continuous with respect to the Lebesgue measure, but it is equivalent to it—meaning that the Lebesgue measure is also absolutely continuous with respect to μ_i.

Definition 2 *The* number of queries *in a protocol is the number of eval and cut queries until termination. We denote the number of queries for a n-player protocol with total demand D by $T(n, D)$.*

The query definition of Woeginger and Sgall is the strictest of the type Robertson-Webb. We now outline three options to extend the notion of a query, all of which have been used in earlier papers [12, 13, 22, 28] and are also referred to as Robertson-Webb queries.

1. **The query definition of Edmonds and Pruhs.** There is a slightly different and stronger formalization of the core idea, given by Edmonds and Pruhs [12] and also used by Procaccia [20, 21]. The crucial difference is that they allow both cut and eval queries to start from an arbitrary point in the cake.

   - *Cut query* (P_i, x, α) returns the leftmost point y so that $\mu_i([x, y]) = \alpha$ or an error message if no such y exists.
   - *Eval query* (P_i, x, y) returns $\mu_i([x, y])$.

   These queries can be simulated as trivial concatenations of the queries defined by Woeginger and Sgall. To pin down the starting point x of a cut query (P_i, x, α) we introduce the cut point x with the help of a dummy player's Lebesgue-measure, ask P_i to evaluate the piece $[0, x]$ and then we cut query with value $\alpha' = \alpha + \mu_i([0, x])$. Similarly, to generate an eval query (P_i, x, y) one only needs to artificially generate the two cut points x and y and then ask two eval queries of the Woeginger-Sgall model, (P_i, x) and (P_i, y). We remark that such a concatenation of Woeginger-Sgall queries reveals more information than the single query in the model of Edmonds and Pruhs.

2. **Proportional cut query.** The term *proportional cut query* stands for n-player Cut queries of the sort "P_i cuts the piece $[x, y]$ in ratio $a : b$", where a, b are integers. As Woeginger and Sgall also note it, two eval queries and one cut query with ratio $\alpha = \frac{a}{a+b} \cdot \mu_i([x, y])$ are sufficient to execute such an operation if x, y are cut points, otherwise five queries suffice. Notice that the eval queries are only used by P_i when she calculates α, and their output does not need to be revealed to any other player or even to the protocol.

3. **Reindexing.** When working with recursive protocols it is especially useful to be able to reindex a piece $[x, y]$ so that it represents the interval $[0, 1]$ for P_i. Any further cut and eval query on $[x, y]$ can also be substituted by at most five queries on the whole cake. Similarly as above, there is no need to reveal the result of the necessary eval queries addressed to a player.

These workarounds ensure that protocols require asymptotically the same number of queries in both model formulations, even if reindexing and proportional queries are allowed. We opted for utilizing all three extensions of the Woeginger-Sgall query model in our upper bound proofs, because the least restrictive model allows the clearest proofs.

2.1 Related work

Equal shares Possibly the most famous cake cutting protocol belongs to the class of Divide and Conquer algorithms. Cut and Choose is a 2-player equal-shares protocol that guarantees proportional shares. It already appeared in the Old Testament, where Abraham divided Canaan to two equally valuable parts and his nephew Lot chose for himself the one he valued more. The first n-player variant of this protocol is attributed to Banach and Knaster in [26] and it requires $\mathcal{O}\left(n^2\right)$ cut and eval queries. Other methods include the continuous (but discretizable) Dubins-Spanier protocol [11] and the Even-Paz protocol [13]. The latter authors show that their method requires $\mathcal{O}\left(n \log n\right)$ queries at most. The complexity of proportional cake cutting has been studied in a setting where a circle instead of an interval represents the cake [3, 5], and also for higher dimensional cakes, where cuts are tailored to fit the shape of the cake [4, 15, 16, 24].

Unequal shares The problem of proportional cake cutting with unequal shares is first mentioned by [26]. Motivated by dividing a leftover cake, Robertson and Webb define the problem formally and offer a range of solutions for two players [22]. More precisely, they list cloning players, using Ramsey partitions [19] and most importantly, the Cut Near-Halves protocol [22]. The last method computes a fair solution for 2 players with integer demands d_1 and d_2 in $2\lceil \log_2(d_1 + d_2) \rceil$ queries. Robertson and Webb also show how any 2-player protocol can be generalized to n players in a recursive manner. The number of physical cuts Cut Near-Halves makes for two players can be beaten for certain demands, as Robertson and Webb also note in [22]. For some demands, Carney [7] and Lohr [17] design such protocols utilizing number-theoretic approaches. Proportional allocation with unequal shares is also discussed in the context of indivisible items instead of a cake [1, 14].

Irrational demands The case of irrational demands in the unequal shares case is interesting from the theoretical point of view, but beyond this, solving it might be necessary, because other protocols might generate instances with irrational demands. For example, in the maximum-efficient envy-free allocation problem with two players and piecewise linear measure functions, any optimal solution must be specified using irrational numbers, as shown in [8]. Barbanel in [2] studies the case of cutting the cake in an irrational ratio between n players and presents a protocol that constructs a proportional division. Shishido and Zeng in [25] solve the same problem with the objective of minimizing the number of resulting pieces. Their protocol is simpler than that of Barbanel [2].

Lower bounds The drive towards establishing lower bounds on the complexity of cake cutting protocols is coeval with the cake cutting literature itself [26]. For proportional cake cutting with equal shares, Even and Paz conjectured that their protocol is the best possible [13], while Robertson and Webb explicitly write that "they would place their money against finding a substantial improvement on the $n \log_2 n$ bound". After approximately 20 years of no breakthrough in the topic, Magdon-Ismail et al. showed in [18] that any protocol must make $\Omega(n \log n)$ comparisons—but this was no bound on the number of queries. Essentially simultaneously, Woeginger and Sgall came up with the lower bound $\Omega(n \log n)$ on the number of queries for the case where contiguous pieces are allocated to each player [28]. Not much later, this condition was dropped by Edmonds and Pruhs [12] who completed the query complexity analysis of proportional cake cutting with equal shares by presenting a lower bound of $\Omega(n \log n)$. Brams et al. [6] study the minimum number of physical cuts in the case of unequal shares and proved that $n-1$ cuts might not suffice—in other words, they show that there in some instances, no proportional allocation exists with contiguous pieces. Crew et al. in [9] and Segal-Halevi in [23] improve this lower bound and shows that at least $2n-2$ cuts may be necessary, and $3n-4$ cuts are always sufficient. However, no lower bound on the number of queries has been known in the case of unequal shares.

3 A protocol for many players with unequal shares

In this section, we present a simple and elegant protocol for cake cutting that beats all three above mentioned protocols (cloning, Ramsey partitions, Cut Near-Halves) in query number. Our main idea is that we recursively render the players in two batches so that these batches can simulate two players who aim to cut the cake into two approximately equal halves. For now we work with the standard cake and

query model defined in Section 2. In what follows, *cutting near-halves* means to cut in ratio $\lfloor \frac{D}{2} \rfloor : \lceil \frac{D}{2} \rceil$.

To ease the notation we assume that the players are indexed so that when they mark the near-half of the cake, the marks appear in an increasing order from 1 to n. In the subsequent rounds, we reindex the players to keep this property intact. Based on these marks, we choose "the middle player" P_j, this being the player whose demand reaches the near-half of the cake when summing up the demands in the order of marks from left to right. This player cuts the cake and each player is ordered to the piece her own mark falls to. The middle player P_j is cloned if necessary so that she can play on both pieces. The protocol is then repeated on both generated subinstances, with adjusted demands. In the subproblem, the players' demands are according to the ratios listed in the pseudocode. The base case of the recursion is a subproblem with one player only, in which case she is allocated the piece.

Proportional division with unequal integer shares

Each player marks the near-half of the cake X.
Sort the players according to their marks.
Calculate the smallest index j such that $\lfloor \frac{D}{2} \rfloor \leq \sum_{i=1}^{j} d_i =: m$.
Cut the cake in two along P_j's mark.
Define two instances of the same problem and solve them recursively.

1. Players $P_1, P_2, \ldots, P_j$ share piece X_1 on the left. Demands are set to $d_1, d_2 \ldots, d_{j-1}, d_j - m + \lfloor \frac{D}{2} \rfloor$, while measure functions are set to $\mu_i \cdot \lfloor \frac{D}{2} \rfloor / \mu_i(X_1)$, for all $1 \leq i \leq j$.

2. Players $P_j, P_{j+1}, \ldots, P_n$ share piece $X_2 = X \setminus X_1$ on the right. Demands are set to $m - \lfloor \frac{D}{2} \rfloor, d_{j+1}, d_{j+2}, \ldots, d_n$, while measure functions are set to $\mu_i \cdot \lceil \frac{D}{2} \rceil / \mu_i(X_2)$, for all $j \leq i \leq n$.

Example 3 *We present our protocol on an example with $n = 3$. Every step of the protocol is depicted in Figure 1. Let $d_1 = 1, d_2 = 3, d_3 = 1$.*

Row 1: Since $D = 5$ is odd, all players mark the near-half of the cake in ratio 2:3. The cake is then cut at P_2's mark, since $d_1 < \lfloor \frac{D}{2} \rfloor$, but $d_1 + d_2 \geq \lfloor \frac{D}{2} \rfloor$.

Row 2: The first subinstance will consist of players P_1 and P_2, both with demand 1, whereas the second subinstance will have the second copy of player P_2 alongside P_3 with demands 2 and 1, respectively. In the first instance, both players mark half of the cake and the one who marked it closer to 0 will receive the leftmost piece, while the other player is allocated the remaining piece. The players in the second instance mark the cake in ratio $1 : 2$. Suppose that the player demanding more marks it closer to 0. The leftmost piece is then allocated to her.

Row 3: The two players in the second subinstance play further: they share the remaining piece in ratio $1 : 1$. The player with the mark on the left will be allocated the piece on the left, while the other players takes the remainder of the piece.

Row 4: The whole cake is divided proportionally.

These rounds require $3 + 2 + 2 + 2 = 9$ proportional cut queries and no eval query.

Theorem 4 *Our "Protocol for proportional division with unequal integer shares" terminates with a proportional division.*

We can estimate the number of queries our protocol needs.

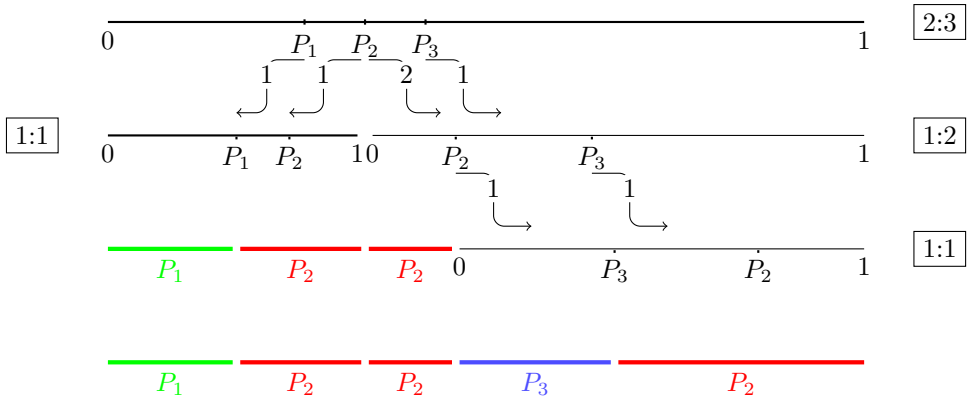


Figure 1: The steps performed by our protocol on Example 3. The colored thick intervals are the pieces already allocated to the player in their label. The ratio players cut the current cake can be seen in the framed labels.

Theorem 5 *For any $2 \leq n$ and $n < D$, the number of queries in our n-player protocol on a cake of total value D is $T(n, D) \leq 2(n-1) \cdot \lceil \log_2 D \rceil$.*

Remark 6 *The "Protocol for proportional division with unequal shares" can be modified to solve the corresponding chore division problem as follows. In the recursive step, players $P_1, P_2, \ldots, P_j$ share piece X_2 rather than X_1 with demands $d_1, d_2, \ldots, d_{j-1}, m - \lfloor \frac{D}{2} \rfloor$ while piece X_1 is shared by players $P_j, P_{j+1}, \ldots, P_n$ with demands $d_j - m + \lfloor \frac{D}{2} \rfloor, d_{j+1}, \ldots, d_n$.*

4 Lower bounds

In this section, we propose an adversary strategy to guarantee that for certain integer demands, any protocol for proportional division requires $\Omega(n \log D)$ queries. This is roughly the same number of queries that our "Proportional division with unequal integer shares" protocol uses. We shall restrict ourselves to the two basic queries described by Woeginger and Sgall in [28], that is to *Cut query* (P_i, α) and *Eval query* (P_i, x). As other queries we mentioned and used earlier can be simulated by a constant number of these two queries, our result remains valid for the wider set of queries, as well.

Our main tool is a nonstandard representation of query-based protocols by two-dimensional segment-vectors. Assume that player P_i has already answered a certain number of queries along some protocol. If the result of *Eval query* (P_i, x_j) is α_j then this information can be stored as a pair (x_j, α_j) such that $\mu_i([0, x_j]) = \alpha_j$. Similarly, if *Cut query* (P_i, α_k) results in x_k then this means that $\mu_i([0, x_k]) = \alpha_k$, and pair (x_k, α_k) stores all the information that this query provides. So all the information that a number of queries reveal can be regarded as a set of (x_j, α_j) pairs and we may assume $x_1 < x_2 < \ldots$. Cutpoints $x_1, x_2, \ldots$ decompose the $[0, 1]$ interval into segments $I_1, I_2, \ldots$ of the cake where $I_j = [x_j, x_{j+1}[$. (For the last segment, we need an artificial cutpoint that belongs to pair $(1, \mu_i([0, 1]))$.) For each such segment I_j, we construct vector $\underline{v}_j = (|I_j|, \mu_i(I_j)) = (x_{j+1} - x_j, \alpha_{j+1} - \alpha_j)$. Let's check how these vectors change after a query!

In case of *Eval query* (P_i, x), there is a j such that $x_j \leq x \leq x_{j+1}$. Hence segment I_j is cut into two new segments $[x_j, x[$ and $[x, x_{j+1}[$. This corresponds to replacing vector $\underline{v}_j$ by vectors $\underline{v'}_j$ and $\underline{v''}_j$ in the segment-vector representation such that $\underline{v}_j = \underline{v'}_j + \underline{v'}_j$.

Assume now that *Cut query* (P_i, α) returns x. Again, there is a j such that $\alpha_1 + \alpha_2 + \ldots + \alpha_j \leq \alpha \leq \alpha_1 + \alpha_2 + \ldots + \alpha_{j+1}$. Consequently, after this particular cut query, segment I_j is cut into two new segments $[x_j, x[$ and $[x, x_{j+1}[$. Again, this corresponds to replacing vector $\underline{v}_j$ by vectors $\underline{v'}_j$ and $\underline{v''}_j$ in the segment-vector representation such that $\underline{v}_j = \underline{v'}_j + \underline{v''}_j$.

These observations show that the segment-vector representation of consecutive queries can be viewed as follows. We start from vector $(1, \mu_i([0,1]))$. At each query, we pick one vector (x_j, α_j) of the representation and specify a number $0 < x < x_j$ or a number $0 < \alpha < \alpha_j$. Depending on which one we specified, player P_i returns a number $0 \leq \alpha \leq \alpha_j$ or a number $0 < x < x_j$. Then vector (x_j, α_j) in the representation is replaced by vectors (x, α) and $(x_j - x, \alpha_j - \alpha)$.

We need one more observation before describing the promised adversary strategy. Assume that a query-based protocol outputs a proportional division for the cake cutting problem and player P_i receives piece X_i. Then $\sum\{\mu_i(I_j) : \lambda(I_j \setminus X_i) = 0\} \geq d_i$ where segments $I_1, I_2, \ldots$ are determined by the queries to P_i. That is, we cannot guarantee more value to P_i than the total value of those segments that are assigned exclusively to P_i (with a possible exception of a null set). The reason for this is that if a positive measure subset Z of some segment I_j is disjoint from X_i then $\mu_i(X_i \cap I_j)$ might be 0. A similar argument shows that in case of proportional chore division, $\sum\{\mu_i(I_j) : \lambda(I_j \cap X_i) > 0\} \leq d_i$ must hold.

Theorem 7 *Assume that $\mu_1([0,1]) = \mu_2([0,1]) = \ldots = \mu_n([0,1]) = 1$, $D \geq n^2$, players $P_0, P_1, \ldots, P_n$ have demands $d_0 = D$, $d_1 = d_2 = \ldots = d_n = 1$ and P_0's valuation is the Lebesgue-measure: $\mu_0 = \lambda$. Then any query-based protocol for the proportional cake cutting problem must use $\Omega(n \log_2 D)$ queries.*

Proof: We describe an adversary strategy that guarantees that if some player $P_1, P_2, \ldots, P_n$ does not answer at least $\Omega(\log_2 D)$ queries then there is no chance to find a proportional division. Along the protocol, each player P_i for $1 \leq i \leq n$ updates the segment-vector representation of her queries and answers any query such that for any $k \geq 0$, after k queries the following property holds for any segment-vector (x, α) of the representation.

$$\alpha \leq \frac{1}{(D+n) \cdot \log_2 D} \text{ or } x \geq 2^{-k} \tag{1}$$

The reader can easily convince herself that players $P_1, P_2, \ldots, P_n$ can answer any query such that property (1) remains true. Assume that after answering k queries, P_i receives piece X_i. By the observation before Theorem 7, $\mu_i(X_i)$ is the sum of the value $\mu_i(I_j)$ of certain segments I_j created after the queries. If $\alpha_j \leq \frac{1}{(D+n)\cdot\log_2 D}$ holds for the all the representing vectors (x_j, α_j) of these I_j's then we need at least $\log_2 D$ segments in order to have $\mu_i(X_i) \geq \frac{1}{D+n}$. This means that P_i had to answer at least $\log_2 D$ queries. Otherwise, X_i contains some segment I_j (with a possible exception of a null set) such that $x_j \geq 2^{-k}$ and hence

$$2^{-k} \leq \lambda(I_j) \leq \lambda(X_i) \leq 1 - \lambda(X_0) \leq \frac{n}{D+n}$$

as P_0 must receive a piece X_0 of size $\lambda(X_0) \geq \frac{D}{D+n}$. This follows that player P_i had to answer at least

$$k \geq \log_2(D+n) - \log_2 n \geq \log_2 D - \log_2 \sqrt{D} = \frac{1}{2} \log_2 D$$

queries, and the theorem follows. □

The chore division version of the above problem seems to be tougher. At least it is not quite clear how the adversary strategy can be modified on order to prove the same lower bound on the number of queries. For two players with demands 1 and D, it is relatively easy to show a $\Omega(\log D)$ lower bound but it is not clear if this can be improved in the multiplayer case. Note that Takács claims a $\Omega(n \log D)$ lower bound for the proportional chore division problem with many players and unequal shares in her BSc dissertation [27].

5 Irrational demands

Contrary to everyday experience where irrational demands may lead to infinite disputes, here we present protocols that find a proportional division after a finite number of queries for both the multiplayer cake

cutting and the chore division problems with unequal and possibly irrational demands. Note that even for two players, there is no upper bound on the number of queries used by our protocols.

Assume that demands $d_1 \leq d_2 \leq \ldots \leq d_n$ of players $P_1, P_2, \ldots, P_n$ are nonnegative reals. We consider a relaxation of the cake-cutting problem: instead of equalities, we require only inequalities $D = \sum_{i=1}^{n} d_i \leq \mu_i([0,1])$ for each $1 \leq i \leq n$. That is, the total demand may be strictly less than the value of the cake, hence it might be possible that $\sum_{i=1}^{n} d_i/\mu_i([0,1]) < 1$ holds for the total share. Definiton 1 is still valid for this relaxation: in a proportional division, every player P_i must receive a piece X_i of $[0,1]$ such that $\mu_i(X_i) \geq d_i$. For chore division, the relaxed condition is $D \geq \mu_i([0,1])$ and Definiton 1 is unchanged. The protocol below outputs a proportional division for the cake-cutting problem.

Proportional division with unequal real shares

1. If $\mu_i([0,1]) > D$ for some i then we pick new demands d'_i for each player P_i such that $d_i \leq d'_i$ and $d'_i/\mu_i([0,1]) \in \mathbb{Q}$ and $\sum_{i=1}^{n} d'_i/\mu_i([0,1]) = 1$. Multiply each d'_i with the common denominator N and apply the "Proportional division with unequal integer shares" protocol with the hence calculated demands $d^*_i = N \cdot d'_i$.

2. If $\mu_i([0,1]) = D \ \forall i$ then each player i marks point *Cut query* (P_i, d_1) on $[0,1]$.

   (a) If the leftmost mark x belongs P_1 then P_1 leaves with piece $X_1 = [0, x[$ and (after reindexing) $P_2, \ldots, P_n$ recursively share part $X_2 = [x, 1]$ with unchanged demands.

   (b) If the mark at x belongs to P_i and not to P_1 then P_i recevives $X_1 = [0, x[$ and $P_1, P_2, \ldots, P_n$ recursively share part $X_2 = [x, 1]$ with demands $d_1, d_2, \ldots, d_{i-1}, d_i - d_1, d_{i+1}, \ldots d_n$.

We prove that the above protocol outputs a proportional division after finitely many queries. In Case 1, we have $\sum_{i=1}^{n} d_i/\mu_i([0,1]) < \sum_{i=1}^{n} d_i/D = \sum_{i=1}^{n} d_i/\sum_{i=1}^{n} d_i = 1$, hence it is possible to pick demands d'_i with the required properties. The output of the called protocoll is a proportional division for demands d^*_i. Hence it is also proportional for demands d'_i, and "even more proportional" for the original demands. Clearly, case 1 requires a finite number of queries.

Observe that in the instance created in Case 2(a) the number of players is decreased by one and $\mu_i(X_2) = D - d_1$ for all i, hence the new task is also a proper instance of the original unrelaxed problem. This shows the correctness of Case 2(a). In case 2(b), no player objects that P_i gets X_1 for the price that her demand is decreased by d_1. Moreover, as $\mu_1(X_2) \geq d - d_1$, the new instance for the recursive call will be in Case 1.

It is rather straightforward to modify above protocol to output a proportional chore division. Namely, the condition for Case 1 is $\mu_i([0,1]) < D$ for some player P_i and we decrease rather than increase the demands in this case. In Case 2, x should be the rightmost mark.

6 Open problems

Here we list two interesting problems that we cannot solve.

1. Is there an irrational number $0 < \alpha < 1$ and a positive integer K such that there exists a protocol that solves the proportional cake cutting problem for two players with demands α and $1 - \alpha$ with no more than K queries?

2. Is there a positive constant c such that our protocol "Proportional division with unequal integer shares" uses at most c times as many queries as any other protocol that finds a proportional division for the cake cutting problem with unequal integer shares?

Acknowledgment

The author thanks Ágnes Cseh for sharing her thoughts on the proportional chore division problem.

References

[1] Moshe Babaioff, Noam Nisan, and Inbal Talgam-Cohen. Competitive equilibrium with indivisible goods and generic budgets. *arXiv preprint arXiv:1703.08150*, 2017.

[2] Julius B Barbanel. Game-theoretic algorithms for fair and strongly fair cake division with entitlements. In *Colloquium Mathematicae*, volume 69:1, pages 59–73, 1996.

[3] Julius B Barbanel, Steven J Brams, and Walter Stromquist. Cutting a pie is not a piece of cake. *The American Mathematical Monthly*, 116(6):496–514, 2009.

[4] Anatole Beck. Constructing a fair border. *The American Mathematical Monthly*, 94(2):157–162, 1987.

[5] Steven J Brams, Michael A Jones, and Christian Klamler. Proportional pie-cutting. *International Journal of Game Theory*, 36(3):353–367, 2008.

[6] Steven J Brams, Michael A Jones, and Christian Klamler. Divide-and-conquer: A proportional, minimal-envy cake-cutting algorithm. *SIAM Review*, 53(2):291–307, 2011.

[7] Edward Carney. A new algorithm for the cake-cutting problem of unequal shares for rational ratios: the divisor reduction method. *Scientific Terrapin*, 3(2):15–22, 2012.

[8] Yuga J Cohler, John K Lai, David C Parkes, and Ariel D Procaccia. Optimal envy-free cake cutting. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, pages 626–631. AAAI Press, 2011.

[9] Logan Crew, Bhargav Narayanan, and Sophie Spirkl. Disproportionate division. *arXiv preprint arXiv:1909.07141*, 2019.

[10] Ágnes Cseh and Tamás Fleiner. The complexity of cake cutting with unequal shares. *ACM Transactions on Algorithms (TALG)*, 16(3):1–21, 2020.

[11] Lester E Dubins and Edwin H Spanier. How to cut a cake fairly. *The American Mathematical Monthly*, 68(1):1–17, 1961.

[12] Jeff Edmonds and Kirk Pruhs. Cake cutting really is not a piece of cake. *ACM Transactions on Algorithms (TALG)*, 7(4):51, 2011.

[13] Shimon Even and Azaria Paz. A note on cake cutting. *Discrete Applied Mathematics*, 7(3):285–296, 1984.

[14] Alireza Farhadi, Mohammad Ghodsi, Mohammad Taghi Hajiaghayi, Sebastien Lahaie, David Pennock, Masoud Seddighin, Saeed Seddighin, and Hadi Yami. Fair allocation of indivisible goods to asymmetric agents. *Journal of Artificial Intelligence Research*, 64:1–20, 2019.

[15] Theodore P Hill. Determining a fair border. *The American Mathematical Monthly*, 90(7):438–442, 1983.

[16] Karthik Iyer and Michael N Huhns. A procedure for the allocation of two-dimensional resources in a multiagent system. *International Journal of Cooperative Information Systems*, 18(03n04):381–422, 2009.

[17] Andrew Lohr. Tight lower bounds for unequal division. *arXiv preprint arXiv:1206.1553*, 2012.

[18] Malik Magdon-Ismail, Costas Busch, and Mukkai S Krishnamoorthy. Cake-cutting is not a piece of cake. In *20th Annual Symposium on Theoretical Aspects of Computer Science*, pages 596–607. Springer Berlin, Heidelberg, 2003.

[19] Kevin McAvaney, Jack Robertson, and William Webb. Ramsey partitions of integers and pair divisions. *Combinatorica*, 12(2):193–201, 1992.

[20] Ariel D Procaccia. Cake cutting: not just child's play. *Communications of the ACM*, 56(7):78–87, 2013.

[21] Ariel D. Procaccia. Cake cutting algorithms. In *Handbook of Computational Social Choice, chapter 13*. Cambridge University Press, 2015.

[22] Jack Robertson and William Webb. *Cake-cutting algorithms: Be fair if you can*. Natick: AK Peters, 1998.

[23] Erel Segal-Halevi. Cake-cutting with different entitlements: How many cuts are needed? *Journal of Mathematical Analysis and Applications*, 480(1):123382, 2019.

[24] Erel Segal-Halevi, Shmuel Nitzan, Avinatan Hassidim, and Yonatan Aumann. Fair and square: Cake-cutting in two dimensions. *Journal of Mathematical Economics*, 70:1–28, 2017.

[25] Harunor Shishido and Dao-Zhi Zeng. Mark-choose-cut algorithms for fair and strongly fair division. *Group Decision and Negotiation*, 8(2):125–137, 1999.

[26] Hugo Steinhaus. The problem of fair division. *Econometrica*, 16:101–104, 1948.

[27] Lili Takács. Igazságos tortaosztási feladat és változatai *Unpublished manuscript*, 2023.

[28] Gerhard J Woeginger and Jiří Sgall. On the complexity of cake cutting. *Discrete Optimization*, 4(2):213–220, 2007.

Algebraic combinatorial optimization for noncommutative rank & determinant

Hiroshi Hirai[1]

Department of Mathematical Informatics,
Graduate School of Information Science and Technology,
The University of Tokyo,
Tokyo, 113-8656, Japan.
hirai@mist.i.u-tokyo.ac.jp

Abstract: Edmonds' problem asks to compute the rank of a linear combination of matrices with variable coefficients:

$$A = \sum_{k=1}^{m} A_k x_k.$$

This problem originates from an algebraic formulation of bipartite matching, and has numerous applications in various areas of mathematical science. A deterministic polynomial time algorithm for Edmonds' problem is not known, and is one of important open problems in theoretical computer science.

Ivanyos, Qiao, and Subrahmanyam introduced a noncommutative formulation of Edmonds' problem, where variables x_k are assumed noncommutative. In this setting, it was shown that the resulting rank (nonncommutative rank; nc-rank) can be computed in deterministic polynomial time. The nc-rank theory is closely linked to combinatorial optimization, and may be viewed as an "algebraization" of it. The present article explains some results on such aspects of the nc-rank and its generalization.

Keywords: Edmonds' problem, noncommutative rank (nc-rank), degree of determinants, fractional matroid matching

1 Introduction

Edmonds' problem [9] asks to compute the rank of a linear combination of matrices with variable coefficients (a *linear symbolic matrix*):

$$A = \sum_{k=1}^{m} A_k x_k, \tag{1.1}$$

where A_k are $n \times n$ matrices over a field $\mathbb{K}$, x_k are variables, and the rank of A is considered in the rational function field $\mathbb{K}(x_1, x_2, \ldots, x_k)$. This problem originates from an algebraic formulation of bipartite matching, and has numerous applications in various areas of mathematical science; see [35]. Although a randomized polynomial time algorithm was shown by Lovász [34], a deterministic polynomial time algorithm for Edmonds' problem is not known, and is one of important open problems in theoretical computer science.

Ivanyos, Qiao, and Subrahmanyam [27] introduced a noncommutative formulation of Edmonds' problem, which regards x_k as noncommutative variables and regards A as a matrix over noncommutative

[1]Research is supported by JSPS KAKENHI Grant Number 21K19759 and JST PRESTO Grant Number JPMJPR192A, Japan.

polynomial ring $\mathbb{K}\langle x_1, x_2, \ldots, x_m\rangle$. The rank of A over the *free skew field* $\mathbb{K}(\langle x_1, x_2, \ldots, x_m\rangle)$ [1, 6] is called the *noncommutative rank (nc-rank)* of A, denoted by nc-rank A. Surprisingly, nc-rank A admits a deterministic polynomial time computation:

Theorem 1.1 ([16, 20, 28]) nc-rank A *of a matrix* A *in* (1.1) *can be computed in polynomial time.*

These polynomial time algorithms are all related to cutting edge technologies of optimization, and have being stimulating subsequent researches. For $\mathbb{K} = \mathbb{Q}$, Garg, Gurvits, Oliveira, and Wigderson [16] showed that Gurvits' *operator scaling* [19] can compute the nc-rank in polynomial time. It turns out that this solves a geodesically-convex optimization problem on a Hadamard manifold; see [3] for further developments. Ivanyos, Qiao, and Subrahmanyam [27, 28] developed a polynomial time algorithm for the nc-rank on an arbitrary field $\mathbb{K}$. Their algorithm (the *Wong sequence algorithm*) is viewed as an algebraic generalization of the classical augmenting-path algorithm for bipartite matching. The algorithm by Hamada and Hirai [20] is a combination of submodular function minimization on a modular lattice and geodesically-convex optimization on a (non-manifold) Hadamard space.

As expected from the origin of Edmonds' problem and the ideas of the last two algorithms, the nc-rank theory is closely linked to combinatorial optimization, and may be viewed as an "algebraization" of it. The present article explains some results on such aspects of the nc-rank and its generalization. The contents of this article is based on [21, 22] and the forthcoming paper [24].

2 Noncommutative rank

The starting point is the formula of nc-rank by Fortin and Reutenauer [10].

Theorem 2.1 ([10]) *Let* A *be a matrix in* (1.1). *Then* nc-rank A *is equal to the optimal value of the following problem:*

$$\begin{array}{lll} \text{(FR)} & \text{Min.} & 2n - r - s \\ & \text{s.t.} & SAT \text{ has an } r \times s \text{ zero submatrix,} \\ & & S, T \in GL_n(\mathbb{K}). \end{array}$$

The problem (FR) is also written as the following vector subspaces optimization (called the *Maximum Vanishing Subspace Problem* in [20])

$$\begin{array}{lll} \text{(MVSP)} & \text{Min.} & 2n - \dim U - \dim V \\ & \text{s.t.} & A_k(U, V) = \{0\} \quad (k \in [m]), \\ & & U, V \subseteq \mathbb{K}^n : \text{vector subspaces,} \end{array}$$

where A_k is regarded as a bilinear form $A_k(x, y) := x^\top A_k y$. It should be noted that this problem already appeared in [35]. MVSP can be viewed as submodular function minimization on the modular lattice of vector subspaces; see [20].

By $\operatorname{rank} A = \operatorname{rank} SAT \leq 2n - r - s$, nc-rank is an upper bound of rank:

$$\operatorname{rank} A \leq \text{nc-rank } A.$$

Theorem 2.2 ([13, 20, 27, 28]) *An optimal solution* S, T *in* FR *can be obtained in polynomial time.*

For $\mathbb{K} = \mathbb{Q}$, the algorithm by Garg et al. [16] can compute the optimal value of FR but cannot obtain an optimal solution (S, T). Recently, Franks, Soma, and Goemans [13] modified this algorithm to obtain an optimal solution (S, T). The algorithm by Hamada and Hirai [20] obtains optimal (S, T) even for small finite fields but does not guarantee a polynomial bit-complexity of (S, T) when $\mathbb{K} = \mathbb{Q}$.

Here we outline the idea of the Wong sequence algorithm by Ivanyos, Qiao, and Subrahmanyam [27, 28]. Let $A = \sum_k A_k x_k$ be a matrix in (1.1). A *substitution* of A is a matrix $\tilde{A}$ over $\mathbb{K}$ obtained from A

by substituting value $z_k \in \mathbb{K}$ to variable x_k for each k, that is, $\tilde{A} = \sum_k A_k z_k$. The *Wong sequence* [26] relative to $(A, \tilde{A})$ is a sequence $W_0, W_1, \ldots,$ of vector subspaces in $\mathbb{K}^n$ defined by

$$W_0 := \{0\}, \quad W_i := \sum_{k=1}^{m} A_k \tilde{A}^{-1} W_{i-1} \quad (i = 1, 2, \ldots), \tag{2.1}$$

where A_k is regarded as $\mathbb{K}^n \to \mathbb{K}^n$.

Lemma 2.3 ([26]) *(1) $W_0 \subset W_1 \subset W_2 \subset \cdots \subset W_j = W_{j+1} = \cdots =: W_\infty$ for some j.*

(2) If $W_\infty \subseteq Im\tilde{A}$, then $\operatorname{rank} \tilde{A} = \operatorname{rank} A = \text{nc-rank}\, A$, *and $(W_\infty^\perp, \tilde{A}^{-1} W_\infty)$ is an optimal solution of MVSP.*

Here $(\cdot)^\perp$ denotes the orthogonal complement. According to this sufficient condition of optimality, the Wong sequence algorithm computes the Wong sequence of a substitution $\tilde{A}$ and its limit W_∞. If $W_\infty \subseteq Im\tilde{A}$, then we obtain an optimal solution of MVSP. Otherwise, the algorithm updates substitution $\tilde{A}$ to increase $\operatorname{rank} \tilde{A}$, or replace A by the blow-up $A^{\{d\}}$ so that nc-rank $A = \frac{1}{d}$ nc-rank $A^{\{d\}}$ (or enlarge ground field $\mathbb{K}$). For a positive integer d, the *d-th blow-up* $A^{\{d\}}$ of A is a linear symbolic matrix defined by

$$A^{\{d\}} := \sum_{k=1}^{m} A_k \otimes X_k, \tag{2.2}$$

where $\otimes$ denotes the Kronecker product and $X_k = (x_{k,ij})$ is a $d \times d$ matrix of variable entries $x_{k,ij}$. We consider the (ordinary) rank of $A^{\{d\}}$ over the rational function field $\mathbb{K}(\{x_{k,ij}\}_{k\in[m],i,j\in[d]})$. The key properties of the blow-up are:

Theorem 2.4 ([27]) (1) nc-rank $A = \max_{d=1,2,\ldots} \frac{1}{d} \operatorname{rank} A^{\{d\}}$.

(2) $\frac{1}{d} \operatorname{rank} A^{\{d\}}$ *is an integer (Regularity Lemma).*

It is known [7] that the maximum in (1) is attained for any $d \geq n - 1$.

In the following, we summarize connections of nc-rank to combinatorial optimization.

Bipartite matching. Let $G = (U \sqcup V, E)$ be a bipartite graph with two color classes U, V and edge set E. Suppose (for simplicity) that $U = V = [n]$. Define a linear symbolic matrix A (*Edmonds' matrix*) by

$$A := \sum_{ij \in E} e_i e_j^\top x_{ij}, \tag{2.3}$$

where e_i denotes the i-th unit vector in $\mathbb{K}^n$. As is well-known, $\operatorname{rank} A$ is equal to the maximum number of a matching of G. It is also equal to nc-rank A. Indeed, matrices S, T in FR can be taken as permutation matrices (vector subspaces U, V in MVSP can be taken as coordinate subspaces). Then nc-rank A is equal to the maximum of $2n - |S|$ over all stable sets S in G. By Kőnig-Egerváry Theorem, this quantity equals the maximum number of a matching. This means $\operatorname{rank} A =$ nc-rank A. In addition, the Wong sequence algorithm (with $0, 1$ substitution) can realize the classical augmenting path algorithm.

Linear matroid intersection. Let $\mathbf{M}_1$ and $\mathbf{M}_2$ be two linearly representable matroids represented by vectors $a_1, a_2, \ldots, a_m$ and $b_1, b_2, \ldots, b_m$ in $\mathbb{K}^n$, respectively. Define a linear symbolic matrix A by

$$A := \sum_{k=1}^{m} a_k b_k^\top x_k. \tag{2.4}$$

Then $\operatorname{rank} A$ is equal to the maximum number of a common independent set of the matroids $\mathbf{M}_1$ and $\mathbf{M}_2$. The feasibility condition of MVSP is: $U^\perp \ni a_k$ or $V^\perp \ni b_k$ for $k \in [m]$. $(U^\perp, V^\perp)$ can be chosen as $(\operatorname{span}\{a_k\}_{k\in I}, \operatorname{span}\{b_k\}_{k\in[m]\setminus I})$ for $I \subseteq [m]$, and the objective value is $r_1(I) + r_2([m] \setminus I)$, where r_i is the rank function of $\mathbf{M}_i$. Namely, nc-rank A equals $\min_{I\subseteq[m]} r_1(I) + r_2([m] \setminus I)$. By the matroid intersection theorem, it equals $\operatorname{rank} A$. Thus, the rank and nc-rank are the same. Further, the Wong sequence algorithm (with $0, 1$ substitution) can realize the matroid intersection algorithm by Edmonds.

Mixed matrix and partitioned matrix. A *mixed matrix* [33] is the sum $Q+T$ of a matrix Q over $\mathbb{K}$ and Edmonds' matrix T for a bipartite graph G, which is also viewed as a linear symbolic matrix $A = Qx_0 + \sum_{ij\in E} E_{ij}x_{ij}$. By Murota's formula [33], its rank is equal to the quantity of MVSP. Thus the rank and nc-rank are the same.

A (generic) *partitioned matrix* in [25] is a linear symbolic matrix of form:

$$A = \begin{pmatrix} A_{11}x_{11} & A_{12}x_{12} & \cdots & A_{1n}x_{1n} \\ A_{21}x_{21} & A_{22}x_{22} & \cdots & A_{2n}x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n1}x_{n1} & A_{n2}x_{n2} & \cdots & A_{nn}x_{nn} \end{pmatrix}, \tag{2.5}$$

where A_{ij} is a matrix (of suitable size) over $\mathbb{K}$ for $i,j \in [n]$.

Iwata and Murota [31] studied the rank of a generic partitioned matrix such that each submatrix A_{ij} is 2×2. They proved a formula of the rank. It turned out that this equals the quantity of MVSP. Hence, the rank and nc-rank are the same. Iwamasa and Hirai [23] showed that the rank equals the maximum of certain algebraically-constraint 2-matchings (*A-matching*) in a bipartite graph associated with A, and sharpened the Wong sequence algorithm to develop a combinatorial polynomial time augmenting path algorithm to compute a maximum A-matching.

Nonbipartite matching and linear matroid matching. A representative example such that rank and nc-rank differ is the Tutte matrix of a nonbipartite graph $G = ([n], E)$. The Tutte matrix of G is a linear symbolic matrix

$$A = \sum_{ij\in E} (e_i e_j^\top - e_j e_i^\top)x_{ij}. \tag{2.6}$$

The maximum matching number of G equals $(1/2)\operatorname{rank} A$. The rank of the Tutte matrix of K_3 is 2, whereas the nc-rank is 3. So the rank and nc-rank differ. Interestingly, the nc-rank still has a natural interpretation: It equals twice the *fractional* matching number of G. This fact was recently revealed by Oki and Soma [37]. They further revealed that this relation is generalized to matroid matching and fractional matroid matching. See Section 3.2.3 for more details. These facts indicate that the nc-rank may be viewed as a "relaxation" of the rank.

3 Degree of Dieudonné determinants

As seen the above examples, the (nc-)rank computation corresponds to cardinality maximization. It is natural to consider an algebraic correspondent of weighed maximization. In fact, this is computation of the degree of determinants. To see this, consider the minimum-weight perfect bipartite matching problem for a bipartite graph $G = ([n] \sqcup [n], E)$ with edge-weight $c : E \to \mathbb{Z}$. Define matrix $A[c]$ by

$$A[c] := \sum_{ij\in E} e_i e_j^\top x_{ij} t^{c(ij)}, \tag{3.1}$$

where t is a new variable commuting with each x_{ij}. Then the (maximum) degree $\deg\det A[c]$ of $\deg A[c]$ with respect to t is equal to the maximum weight of a perfect matching in G. For a matrix A of linear matroid intersection (2.4), if $A[c]$ is defined similarly, then $\deg\det A[c]$ is equal to the maximum weight of a common base of $\mathbf{M}_1$ and $\mathbf{M}_2$ with rank n.

3.1 Linear symbolic rational matrices

Motivated by this observation, Hirai [21] formulated a noncommutative version of the deg-det computation. Consider a general class of linear symbolic rational matrices:

$$B = B_1x_1 + B_2x_2 + \cdots + B_mx_m, \tag{3.2}$$

where x_k are variables as above, and $B_k = B_k(t)$ are $n \times n$ matrices over rational function field $\mathbb{K}(t)$ with indeterminate t. Then $\deg\det B$ is considered in rational function field $\mathbb{K}(x_1, x_2, \ldots, x_m, t)$.

The noncommutative formulation is as follows: see [21, 22, 36] for details. The matrix B is regarded as the rational function skew field $\mathbb{F}(t) = \{p/q \mid p \in \mathbb{F}[t], q \in \mathbb{F}[t] \setminus \{0\}\}$, where $\mathbb{F} := \mathbb{K}(\langle x_1, \ldots, x_m\rangle)$ is the skew free field, $\mathbb{F}[t]$ is the skew polynomial ring over $\mathbb{F}$, and the degree of an element in $\mathbb{F}(t)$ is defined similarly. Any (nonsingular) matrix B over $\mathbb{F}(t)$ is written as $B = LDPU$ (called the *Bruhat normal form*), where L and U are lower-triangular and upper-triangular, respectively, matrices with unit diagonals, P is a permutation matrix, and D is a diagonal matrix. The *Dieudonné determinant* $\mathrm{Det}\, B$ of B is defined as the product of the sign of P and all diagonals of D modulo the commutator subgroup of the multiplicative group $\mathbb{F}(t) \setminus \{0\}$ [8]. We let $\mathrm{Det}\, B := 0$ if B is singular. Since DP is uniquely determined, $\mathrm{Det}\, B$ is well-defined. Although $\mathrm{Det}\, B$ is no longer an element of $\mathbb{F}(t)$, its degree $\deg \mathrm{Det}\, B \in \mathbb{Z} \cup \{-\infty\}$ is well-defined (as the degree of any commutator is zero).

In addition to $\deg\det$ and $\deg \mathrm{Det}$, we consider the maximum degrees of subdeterminants of the two types: For $\ell \in [n]$, define

$$\delta_\ell(B) := \max\{\deg\det B[I,J] \mid I, J \subseteq [n] : |I| = |J| = \ell\}, \tag{3.3}$$

$$\Delta_\ell(B) := \max\{\deg \mathrm{Det}\, B[I,J] \mid I, J \subseteq [n] : |I| = |J| = \ell\}, \tag{3.4}$$

where $B[I,J]$ denote the submatrix of B having row set I and column set J.

3.1.1 Duality theorem

As an extension of the formula of nc-rank (Theorem 2.1), Hirai [21] established the following formula of $\deg \mathrm{Det}$:

Theorem 3.1 ([21]) *Let $B = B(t)$ be a matrix in* (3.2)*. Then* $\deg \mathrm{Det}\, B$ *is equal to the optimal value of the following problem (Maximum Vanishing subModule Problem):*

$$\begin{array}{lll} \text{(MVMP)} & \text{Min.} & -\deg\det P - \deg\det Q \\ & \text{s.t.} & \deg(PB_kQ)_{ij} \le 0 \quad (i,j \in [n], k \in [m]), \\ & & P, Q \in GL_n(\mathbb{K}(t)). \end{array}$$

This problem can be viewed as an *L-convex function* minimization on the lattice of certain submodules in $\mathbb{K}(t)^n$ [21]. It should be noted that the quantity of MVMP already appeared in [32] as an upper bound of $\deg\det$. In particular,

$$\deg\det B \le \deg \mathrm{Det}\, B.$$

Any matrix $B = B(t)$ over $\mathbb{K}(t)$ is written as a formal power series as $B = B^{(d)}t^d + B^{(d-1)}t^{d-1} + \cdots$, where $B^{(\ell)}$ is a matrix over $\mathbb{K}$ ($\ell = d, d-1, \ldots$) and $d \ge \max_{ij} \deg B_{ij}$. The feasibility condition of MVMP is rephrased as: PBQ is written as $PBQ = (PBQ)^{(0)} + (PBQ)^{(-1)}t^{-1} + \cdots$. The linear symbolic matrix of the leading term $(PBQ)^{(0)} = \sum_k (PB_kQ)^{(0)}x_k$ plays particularly important roles.

Lemma 3.2 ([21]) *Let (P,Q) be a feasible solution for MVMP.*

(1) *(P,Q) is optimal if and only if* $\text{nc-rank}(PBQ)^{(0)} = n$.

(2) *If* $\mathrm{rank}(PBQ)^{(0)} = n$, *then* $\deg\det B = \deg \mathrm{Det}\, B = -\deg\det P - \deg\det Q$.

3.1.2 Deg-Det algorithm

We here explain the **Deg-Det** algorithm [21] to solve MVMP. This algorithm uses an algorithm of solving FR as a subroutine, and is viewed as a simplified version of Murota's *combinatorial relaxation algorithm* [32] developed for $\deg\det$; see also [33, Section 7.1].

We assume that the position of a zero submatrix in FR is upper right. For $s \in [n] := \{1, 2, \ldots, n\}$, let $\mathbf{1}_s := \sum_{i=1}^{s} e_i \in \mathbb{Z}^n$. For integer vector $\alpha \in \mathbb{Z}^n$, let (t^α) denote the diagonal matrix with diagonals $t^{\alpha_1}, t^{\alpha_2}, \ldots, t^{\alpha_n}$ in order.

Algorithm: Deg-Det

Input: $B = \sum_{k=1}^{m} B_k x_k$, and a feasible solution P, Q for MVMP.

Output: $\deg \mathrm{Det}\, B$.

1: Solve the problem FR for $(PBQ)^{(0)}$ and obtain optimal matrices S, T.

2: If the optimal value $2n - r - s$ of FR is equal to n, then output $-\deg \det P - \deg \det Q$. Otherwise, letting $(P, Q) \leftarrow ((t^{\mathbf{1}_r})SP, QT(t^{-\mathbf{1}_{n-s}}))$, go to step 1.

The algorithm works as follows: The matrix $SPAQT$ after step 1 has a negative degree in each entry of its upper right $r \times s$ submatrix. Multiplying t for the first r rows and t^{-1} for the first $n - s$ columns yields no entry of positive degree. Thus, the next solution $(P, Q) := ((t^{\mathbf{1}_r})SP, QT(t^{-\mathbf{1}_{n-s}}))$ is feasible, and decreases the objective value by $r + s - n (> 0)$. If the algorithm terminates, then (P, Q) is optimal by Lemma 3.2 (1). If the algorithm does not terminate, then $\deg \mathrm{Det}\, B = -\infty$.

In step 2, we can replace $(\mathbf{1}_r, \mathbf{1}_{n-s})$ by $(\alpha \mathbf{1}_r, \alpha \mathbf{1}_{n-s})$ for the maximum possible integer $\alpha > 0$ so that the next solution is feasible. In the bipartite instance (3.1), the **Deg-Det** algorithm with this update can simulate the classical Hungarian method. Also, for matroid intersection instance, Furue and Hirai [14] showed that the **Deg-Det** algorithm can derive Frank's *weighted splitting algorithm* [11] with a new matrix implementation. In the both cases, the leading matrix $(PBQ)^{(0)}$ in step 2 can keep the rank-1 property, and Lemma 3.2 (2) shows $\deg \det = \deg \mathrm{Det}$. The same approach proves $\deg \det = \deg \mathrm{Det}$ for 2×2-partitioned matrices [22].

Consider computation of maximum subdeterminants $\Delta_\ell(B)$. It is known that $(I, J) \mapsto \deg \det B[I, J]$ is a valuated bimatroid; see [33]. The same holds for $\deg \mathrm{Det}\, B[I, J]$.

Theorem 3.3 ([21]) *$(I, J) \mapsto \deg \mathrm{Det}\, B[I, J]$ is a valuated bimatroid.*

Therefore, by the incremental greedy algorithm [33, 5.2.5], the whole $\Delta_\ell(B)$ can be computed by a polynomial number of calls of computation of $\deg \mathrm{Det}\, B[I, J]$.

3.2 Linear symbolic monomial matrices

We restrict ourselves to a special class of linear symbolic rational matrices. For a matrix $A = \sum_{k=1}^{m} A_k x_k$ in (1.1) and $c \in \mathbb{Z}^m$, define a linear symbolic rational matrix $A[c]$ by

$$A[c] := \sum_{k=1}^{m} A_k x_k t^{c_k}. \tag{3.5}$$

As seen above, this class of linear symbolic matrices captures weighted maximization of several combinatorial optimization problems. The **Deg-Det** algorithm is applicable for $A[c]$ but is a pseudo polynomial in c. By regarding c as a cost vector, Hirai and Ikeda [22] incorporated *cost scaling* with the **Deg-Det** algorithm, and obtained a polynomial time algorithm for $\deg \mathrm{Det}\, A[c]$:

Theorem 3.4 ([22]) *Let A be a matrix in (1.1) and let $c \in \mathbb{Z}^m$.*

(1) *Suppose that arithmetic operations over $\mathbb{K}$ are performed in constant time. Then $\deg \mathrm{Det}\, A[c]$ can be computed in time polynomial of $n, m, \log C$, where $C := \max_k |c_k|$.*

(2) *Suppose that $\mathbb{K} = \mathbb{Q}$ and that each A_k consists of integer entries whose absolute values are at most $D > 0$. Then $\deg \mathrm{Det}\, A[c]$ can be computed in time polynomial of $n, m, \log C, \log D$.*

They also utilized the Frank-Tardos method [12] to remove $\log C$ from the complexity. The second result (2) is based on a polyhedral interpretation of $\deg \mathrm{Det}$ (Section 3.2.2) and the modulo-p reduction method by Iwata and Kobayashi [30] devised for the weighted linear matroid matching problem. Specifically, $\deg \mathrm{Det}\, A[c]$ is equal to the maximum of $\deg \mathrm{Det}(A \mod p)[c]$ for a polynomial number of primes p, where the bit-length of p is also polynomially bounded and $A \mod p$ is a linear symbolic matrix over finite field $GF(p)$.

3.2.1 Duality theorem

The forthcoming paper [24] sharpens Theorem 3.1 for $\Delta_\ell(A[c])$ as follows:

Theorem 3.5 ([24]) *Let $A = \sum_{k=1}^m A_k x_k$ be a matrix in (1.1) and let $c \in \mathbb{Z}^m$. Then $\Delta_\ell(A[c])$ is equal to the optimal value of the following problem:*

$$\begin{aligned}
\text{Min.} \quad & -\sum_{i=n-\ell+1}^{n} \alpha_i - \sum_{j=n-\ell+1}^{n} \beta_j \\
\text{s.t.} \quad & \alpha_i + \beta_j \le -c_k \quad (i,j \in [n], k \in [m] : A_k(U_i, V_j) \neq \{0\}), \\
& \alpha_1 \ge \alpha_2 \ge \cdots \ge \alpha_n, \ \beta_1 \ge \beta_2 \ge \cdots \ge \beta_n, \\
& U_1 \subset U_2 \subset \cdots \subset U_n, \ V_1 \subset V_2 \subset \cdots \subset V_n, \\
& \alpha, \beta \in \mathbb{Z}^n, U_i, V_j \subseteq \mathbb{K}^n : \textit{vector subspaces for } i,j \in [n].
\end{aligned}$$

This problem is also written as

$$\begin{aligned}
\text{Min.} \quad & \sum_{i=1}^{n} p_i + \sum_{j=1}^{n} q_j + \ell\gamma \\
\text{s.t.} \quad & p_i + q_j + \gamma \ge c_k \quad (i,j \in [n], k \in [m] : A_k(u_i, v_j) \neq \{0\}), \\
& \{u_1, u_2, \ldots, u_n\}, \{v_1, v_2, \ldots, v_n\} : \text{bases of } \mathbb{K}^n, \\
& p, q \in \mathbb{Z}_+^n, \gamma \in \mathbb{Z}.
\end{aligned}$$

Observe that this is LP-dual of the weighted bipartite matching problem if the bases are fixed, and provides a good characterization for $\Delta_\ell(A[c])$ if the bit-length of the bases is bounded.

For the case of a 2×2-partitioned matrix A, Iwamasa [29] developed a primal-dual combinatorial polynomial time algorithm to compute $\delta_\ell(A[c]) = \Delta_\ell(A[c])$, and provided an algorithmic proof of the above duality.

3.2.2 Polyhedral interpretation

Here we explain a polyhedral interpretation of $\deg \mathrm{Det}$. Before that, we first consider $\deg \det$. For a multivariate polynomial $p(x_1, x_2, \ldots, x_m) = \sum_{u_1,u_2,\ldots,u_m} a_{u_1 u_2 \ldots u_m} x_1^{u_1} x_2^{u_2} \cdots x_m^{u_m}$, let $\mathrm{vec}\, p \subseteq \mathbb{Z}^m$ denote the set of all integer vectors $u = (u_1, u_2, \ldots, u_m)$ with $a_{u_1 u_2 \ldots u_m} \neq 0$. Let $P_\ell(A) \subseteq \mathbb{R}^m$ be the polytope defined by

$$P_\ell(A) := \mathrm{Conv} \bigcup \{\mathrm{vec}\det A[I,J] \mid I, J \subseteq [n] : |I| = |J| = \ell\}. \tag{3.6}$$

The maximum degrees of subdeterminants of $A[c]$ are given by linear optimizations over $P_\ell(A)$:

Lemma 3.6 $\delta_\ell(A[c]) = \max\{c^\top u \mid u \in P_\ell(A)\}$.

Hirai and Ikeda [22] revealed an analogous interpretation for $\deg \mathrm{Det}\, A[c]$ (and thus for $\Delta_\ell(A[c])$). Recall the d-th blow up $A^{\{d\}}$ of A, and consider its ordinary determinant $\det A^{\{d\}}$ that is a polynomial of variables $x_{k,ij}$ for $k \in [m], i,j \in [n]$. The exponent vector $\mathrm{vec}\det A^{\{d\}}$ is an md^2-dimensional integer vector $z = (z_{k,ij})_{k\in[m],i,j\in[d]}$. For such vector $z = (z_{k,ij})_{k\in[m],i,j\in[d]}$, define an m-dimensional vector $\mathrm{proj}_d(z) \in \mathbb{Q}^m$ by

$$\mathrm{proj}_d(z)_k := \frac{1}{d} \sum_{i,j\in[d]} z_{k,ij} \quad (k \in [m]). \tag{3.7}$$

Define $Q_\ell(A) \subseteq \mathbb{R}^m$ by

$$Q_\ell(A) := \mathrm{Conv} \bigcup \{\mathrm{proj}_d \,\mathrm{vec}\det A[I,J]^{\{d\}} \mid I, J \subseteq [n] : |I| = |J| = \ell, d = 1, 2, \ldots\}. \tag{3.8}$$

An analogue of Theorem 2.4 and Lemma 3.6 is the following.

Theorem 3.7 ([24, 22]) (1) $\Delta_\ell(A[c]) = \max\{c^\top u \mid u \in Q_\ell(A)\}$.

(2) *$Q_\ell(A)$ is an integral polytope belonging to $\{u \in \mathbb{R}^m \mid \mathbf{1}^\top u = \ell\}$.*

(3) *An integral vector u maximizing $c^\top u$ over $Q_\ell(A)$ is obtained in polynomial time.*

The meaning of polynomiality in (3) is the same as in Theorem 3.4. In particular, $Q_\ell(A)$ is an integral relaxation of $P_\ell(A)$. For a matrix A of linear matroid intersection, $Q_\ell(A)$ and $P_\ell(A)$ are equal, and the vertices of $Q_\ell(A) = P_\ell(A)$ are incidence vectors of common independent sets of cardinality ℓ. The same holds for 2×2-partitioned matrices A, in which the vertices of $Q_n(A)$ is the incidence vectors of perfect A-matchings [22].

3.2.3 Fractional linear matroid matching and rank-2 Brascamp-Lieb polytope

Let $\mathcal{H} = \{H_1, H_2, \ldots, H_m\}$ be a collection of 2-dimensional subspaces in $\mathbb{K}^n$. A *fractional matroid matching* for $\mathcal{H}$ (Vande Vate [38]) is a nonnegative vector $y \in \mathbb{R}^m_+$ satisfying

$$\sum_{k=1}^{m} y_k \dim H_k \cap X \leq \dim X \quad (X \subseteq \mathbb{K}^n : \text{vector subspace}).$$

In addition, if $2\sum_{k=1}^m y_k = n$, then it is called *perfect*. The *fractional matroid matching polytope* $FMP(\mathcal{H})$ for $\mathcal{H}$ is the polytope consisting of all fractional matroid matchings. Suppose that $H_k = \mathrm{span}\{a_k, b_k\}$ for $a_k, b_k \in \mathbb{K}^n, k \in [m]$. Define linear symbolic matrix A by

$$A := \sum_{k=1}^{m} (a_k b_k^\top - b_k a_k^\top) x_k. \tag{3.9}$$

It is known [35] that $\mathrm{rank}\, A$ is equal to twice the maximum number of a matroid matching (a subset $I \subseteq [m]$ with $\dim \sum_{k\in I} H_k = 2|I|$). Oki and Soma [37] showed that nc-$\mathrm{rank}\, A$ is equal to twice the maximum size of a fractional matroid matching.

Theorem 3.8 ([37]) nc-$\mathrm{rank}\, A = 2\max\{\mathbf{1}^\top y \mid y \in FMP(\mathcal{H})\}$.

They also showed that the second blow-up $A^{\{2\}}$ attains the nc-rank, and developed a fast randomized algorithm for solving the fractional matching matroid problem.

We extend the above relation to a weighted version.

Theorem 3.9 ([24]) $\Delta_\ell(A[c]) = 2\max\{c^\top y \mid y \in FMP(\mathcal{H}), 2\mathbf{1}^\top y = \ell\}$.

Thus, the above framework for $\deg \mathrm{Det}$ is applicable to the weighted fractional matching problem. This relation can be shown by interpreting the problem in (3.5) as LP-dual of linear optimization over $FMP(\mathcal{H})$. Particularly, $Q_n(A)$ is equal to twice the polytope $PFMP(\mathcal{H})$ of perfect fractional matroid matchings.

Franks, Soma, and Goemans [13] revealed an interesting connection between $PFMP(\mathcal{H})$ and the *Brascamp-Lieb inequality*. It is known [2] that the finiteness of the constant of Brascamp-Lieb inequality can be decided by solving the membership problem of a certain polytope, called the *Brascamp-Lieb polytope (BL-polytope)*. The computational complexity of the BL-polytope has attracted attention in theoretical computer science [15]. Franks, Soma, and Goemans [13] showed that when the BL-inequality is associated with rank-2 matrices $B_1, B_2, \ldots, B_m$, the corresponding BL-polytope (*rank-2 BL-polytope*) coincides with $PFMP(\mathcal{H})$ for 2-dimensional spaces spanned by B_k. By general principle of optimization and separation [18], a polynomial time algorithm for linear optimization over $PFMP(\mathcal{H})$ (strong optimization oracle) implies polynomial complexity of the strong separation, particularly, the membership of $PFMP(\mathcal{H})$. Gijswijt and Pap [17] reduced the weighted fractional matroid matching problem to a polynomial number of unweighted problems for which a polynomial time algorithm is given by Chang, Llewellyn, and Vande Vate [4, 5]. However, this reduction is not enough, since it can cause bit-explosion

for $\mathbb{K} = \mathbb{Q}$, the setting of the BL-inequality, as pointed out by [13]. They showed via modified operator scaling that the membership of $PFMP(\mathcal{H})$ is in NP $\cap$ coNP for $\mathbb{K} = \mathbb{Q}$.

Theorem 3.7 (3) and Theorem 3.9 imply polynomial-time solvability of linear optimization over $PFMP(\mathcal{H})$, even for $\mathbb{K} = \mathbb{Q}$. Thus:

Theorem 3.10 ([24]) *The strong separation problem in the rank-2 BL-polytope is in* P.

References

[1] S. A. Amitsur: Rational identities and applications to algebra and geometry. *Journal of Algebra* **3** (1966), 304–359.

[2] J. Bennett, A. Carbery, M. Christ, and T. Tao: The Brascamp-Lieb inequalities: Finiteness, structure and extremals. *Geometric and Functional Analysis* **17**(2008), 1343–1415.

[3] P. Bürgisser, C. Franks, A. Garg, R. Oliveira, M. Walter, and A. Wigderson: Towards a theory of non-commutative optimization: geodesic first and second order methods for moment maps and polytopes. preprint, 2019. (the conference version in FOCS 2019)

[4] S. Chang, D. C. Llewellyn, and J. H. Vande Vate: Matching 2-lattice polyhedra: finding a maximum vector. *Discrete Mathematics* **237** (2001), 29–61.

[5] S. Chang, D. C. Llewellyn, and J. H. Vande Vate: Two-lattice polyhedra: duality and extreme points. *Discrete Mathematics* **237** (2001), 63–95.

[6] P. M. Cohn: *Skew Fields: Theory of General Division Rings.* Cambridge University Press, Cambridge, 1995.

[7] H. Derksen and V. Makam: Polynomial degree bounds for matrix semi-invariants. *Advances in Mathematics* **310** (2017) 44–63.

[8] J. Dieudonné: Les déterminants sur un corps non commutatif. *Bulletin de la Société Mathématique de France* **71** (1943) 27–45.

[9] J. Edmonds: Systems of distinct representatives and linear algebra. *Journal of Research of the National Bureau of Standards* **71B** (1967) 241–245.

[10] M. Fortin and C. Reutenauer: Commutative/non-commutative rank of linear matrices and subspaces of matrices of low rank. *Séminaire Lotharingien de Combinatoire* **52** (2004), B52f.

[11] A. Frank: A weighted matroid intersection algorithm. *Journal of Algorithms* **2** (1981), 328–336.

[12] A. Frank and É. Tardos: An application of simultaneous Diophantine approximation in combinatorial optimization. *Combinatorica* **7** (1987), 49–65.

[13] C. Franks, T. Soma, and M. Goemans: Shrunk subspaces via operator Sinkhorn iteration. preprint, 2022 (the conference version in SODA 2023).

[14] H. Furue and H. Hirai: On a weighted linear matroid intersection algorithm by deg-det computation. *Japan Journal of Industrial and Applied Mathematics* **37** (2020), 677–696.

[15] A. Garg, L. Gurvits, R. Oliveira, and A. Wigderson: Algorithmic and optimization aspects of Brascamp–Lieb inequalities, via operator scaling. *Geometric and Functional Analysis* **28** (2018), 100–145. (the conference version in STOC 2017)

[16] A. Garg, L. Gurvits, R. Oliveira, and A. Wigderson: Operator scaling: theory and applications. *Foundations of Computational Mathematics* **20** (2020), 223–290. (the conference version in FOCS 2016)

[17] D. Gijswijt and G. Pap: An algorithm for weighted fractional matroid matching. *Journal of Combinatorial Theory, Series B* **103** (2013), 509–520.

[18] M. Grötchel, L. Lovász, and A. Schrijver: *Geometric Algorithms and Combinatorial Optimization*. Springer-Verlag, Berlin, 1993.

[19] L. Gurvits: Classical complexity and quantum entanglement. *Journal of Computer and System Sciences* **69** (2004), 448–484.

[20] M. Hamada and H. Hirai: Computing the nc-rank via discrete convex optimization on CAT(0) spaces. *SIAM Journal on Applied Geometry and Algebra* **5** (2021), 455–478. (the conference version in JH 2017)

[21] H. Hirai: Computing the degree of determinants via discrete convex optimization on Euclidean buildings. *SIAM Journal on Applied Geometry and Algebra* **3** (2019), 523–557.

[22] H. Hirai and M. Ikeda: A cost-scaling algorithm for computing the degree of determinants. *Computational Complexity* **31** (2022) Article number: 10.

[23] H. Hirai and Y. Iwamasa: A combinatorial algorithm for computing the rank of a generic partitioned matrix with 2×2 submatrices. *Mathematical Programming, Series A* **195** (2022), 1–37. (the conference version in IPCO 2020)

[24] H. Hirai, Y. Iwamasa, T. Oki, and T. Soma: An improved analysis on deg-Det computation of symbolic matrices and its applications. in preparation.

[25] H. Ito, S. Iwata, and K. Murota: Block-triangularizations of partitioned matrices under similarity/equivalence transformations. *SIAM Journal on Matrix Analysis and Applications* **15** (1994), 1226–1255.

[26] G. Ivanyos, M. Karpinski, Y. Qiao, and M. Santha: Generalized Wong sequences and their applications to Edmonds' problems. *Journal of Computer and System Sciences* **81** (2015) 1373–1386.

[27] G. Ivanyos, Y. Qiao, and K. V. Subrahmanyam: Non-commutative Edmonds' problem and matrix semi-invariants. *Computational Complexity* **26** (2017), 717–763.

[28] G. Ivanyos, Y. Qiao, and K. V. Subrahmanyam: Constructive noncommutative rank computation in deterministic polynomial time over fields of arbitrary characteristics. *Computational Complexity* **27** (2018), 561–593. (the conference version in ITCS 2017)

[29] Y. Iwamasa: A combinatorial algorithm for computing the entire sequence of the maximum degree of minors of a generic partitioned polynomial matrix with 2×2 submatrices. preprint, (2021) (the conference version in IPCO 2021).

[30] S. Iwata and Y. Kobayashi: A weighted linear matroid parity algorithm. *SIAM Journal on Computing* **51** (2022), 238–280. (the conference version in STOC 2017).

[31] S. Iwata and K. Murota: A minimax theorem and a Dulmage-Mendelsohn type decomposition for a class of generic partitioned matrices. *SIAM Journal on Matrix Analysis and Applications* **16** (1995), 719–734.

[32] K. Murota: Computing the degree of determinants via combinatorial relaxation. *SIAM Journal on Computing* **24** (1995), 765–796.

[33] K. Murota: *Matrices and Matroids for Systems Analysis*. Springer-Verlag, Berlin, 2000.

[34] L. Lovász: On determinants, matchings, and random algorithms. In:*Fundamentals of Computation Theory FCT'79 Proceedings of Algebraic, Arithmetic. and Categorical Methods in Computation Theory* (L. Budach, et.), Akademie-Verlag, Berlin (1979) 565–574.

[35] L. Lovász: Singular spaces of matrices and their application in combinatorics. *Boletim da Sociedade Brasileira de Matemática* **20** (1989), 87–99.

[36] T. Oki: Computing valuations of the Dieudonné determinants. *Journal of Symbolic Computation* **116** (2023), 284–323.

[37] T. Oki and T. Soma: Algebraic algorithms for fractional linear matroid parity via non-commutative rank, preprint, (2022). (the conference version in SODA 2023)

[38] J. H. Vande Vate: Fractional matroid matchings. *Journal of Combinatorial Theory, Series B* **55** (1992), 133–145.

Matching in Bipartite Graphs with Stochastic Arrivals and Departures[1]

Naonori Kakimura

Department of Mathematics
Keio University
Yokohama, Japan
kakimura@math.keio.ac.jp

Donghao Zhu

Technical University of Munich,
Munich, Germany
donghao.zhu@in.tum.de

Abstract: In this paper, we study a matching market model on a bipartite network where agents on each side arrive and depart stochastically by a Poisson process. For such a dynamic model, we design a mechanism that decides not only which agents to match, but also when to match them, to minimize the expected number of unmatched agents. The main contribution of this paper is to achieve theoretical bounds on the performance of local mechanisms with different timing properties. We show that an algorithm that waits to thicken the market, called the *Patient* algorithm, is exponentially better than the *Greedy* algorithm, i.e., an algorithm that matches agents greedily. This means that waiting has substantial benefits on maximizing a matching over a bipartite network. We remark that the Patient algorithm requires the planner to identify agents who are about to leave the market, and, under the requirement, the Patient algorithm is shown to be an optimal algorithm. We also show that, without the requirement, the Greedy algorithm is almost optimal. In addition, we consider the *1-sided algorithms* where only an agent on one side can attempt to match. This models a practical matching market such as a freight exchange market and a labor market where only agents on one side can make a decision. For this setting, we prove that the Greedy and Patient algorithms admit the same performance, that is, waiting to thicken the market is not valuable. This conclusion is in contrast to the case where agents on both sides can make a decision and the non-bipartite case by [Akbarpour et al., *Journal of Political Economy*, 2020].

Keywords: Bipartite matching, Markov chain, Random graph, Online algorithm.

1 Introduction

Matching markets arise in many applications such as marriage and dating market [10], paired kidney exchange [2], and ride-hailing system [3, 13]. In a matching market, which can be modeled as a network with agents (vertices) and edges, a social planner designs a mechanism that finds an acceptable matching on the network. In a *dynamic* matching market, agents are allowed to arrive and depart over time. A market is then changed dynamically over time, in which a social planner designs a mechanism that chooses how to match agents.

A dynamic matching market has been studied extensively in theory [9, 11, 2] and practice [5, 6]. Recently, Akbarpour et al. [2] introduced a seminal matching market model with arrivals and departures. In their model, agents arrive at and depart from the market according to the Poisson process. The planner observes the network and chooses a matching, aiming to minimize the number of unmatched agents. One of the key feature in their model is that the planner must decide not only which agents to match, but also *when* to match them. Akbarpour et al. [2] showed that the choice of when to match agents has large effects on performance. Specifically, they introduced two simple mechanisms with different timing

[1]The full version of this paper is available at [8]. Research is supported by 21H03397, 20H05795, and 22H05001.

properties, *Greedy* and *Patient*. They provided theoretical guarantees for these mechanisms, that suggests waiting has substantial benefits on maximizing a matching over the network.

This paper focuses on a *bipartite matching market* where the network is a bipartite graph. Agents in the market are divided into two separated groups, and a matching is formed between the two groups. A bipartite matching market is one of the most popular matching markets in practice; a labor market matches a worker to a task, and a ride-hailing market matches a taxi to a passenger [4, 12].

We propose a bipartite matching market model with arrivals and departures as a variant of Akbarpour et al.'s (non-bipartite) matching model. We aim at designing *local* algorithms in the sense that they look only at the neighbors of an agent which attempts to match, rather than at the global network structure. Local algorithms can be viewed as a mechanism that each agent individually decides to find a partner. In a bipartite matching market, agents in two separated groups have different roles, and agents on one side often have no right to make a decision. For example, in a freight exchange market between shippers and carriers, some platforms such as Wtransnet only allow carriers to choose shipments. For another, in a competitive labor market, only workers submit job applications to companies, and companies make final decisions. Thus, it is natural to consider the situation when agents on only one side have a right to make a decision. Such a setting is called a *1-sided market* [1]. We also consider the situation when agents on both sides can make a decision, called a *2-sided market*, that also appears in practice such as a marriage and dating market and a freight exchange market like Cargopedia.

2 Our Contributions

In this work, we evaluate the performance of simple local mechanisms on a bipartite matching market to measure the impact of waiting time in the 1-sided/2-sided markets. Our main contributions are summarized as follows:

- We introduce a formal framework of 1-sided/2-sided bipartite market model with arrivals and departures. We propose algorithms with different timing properties, Greedy and Patient algorithms, for the 1-sided and 2-sided markets, respectively. We present almost optimal bounds on the performance of these algorithms. Our results show that waiting to thicken the market is highly valuable for the 2-sided market, while it is not true for the 1-sided market.

- We provide lower bounds on the performance of any matching algorithms. We show that, if the planner does not know the information when an agent departs, any algorithm suffers a loss exponentially larger than that of an omniscient algorithm where the information is available.

Let us describe our results in more detail.

Model In our model, agents in two classes arrive at Poisson rates λ_a and λ_b, respectively, and a pair of two agents in different classes are compatible with probability p. Each agent departs at a Poisson rate, normalized to 1. The planner chooses a matching on the current network, and matched agents leave the market. The planner aims to minimize the proportion of the expected unmatched agents (called *the loss*). This setting is a variant of a matching market model by Akbarpour et al. [2], where each agent arrives at Poisson rate λ and edges are formed between any pair of agents with probability p.

In this paper, we consider two simple mechanisms, *Greedy* and *Patient*, for a bipartite matching market. The Greedy algorithm attempts to match an agent upon her arrival, while the Patient algorithm attempts to match only an urgent agent, that is, an agent at departure. Note that both these algorithms are local, in the sense that an agent individually makes a decision when she arrives in the Greedy algorithm, and when she departs in the Patient algorithm. In the 2-sided market, every agent attempts to match to some agent according to Greedy or Patient algorithms, while, in the 1-sided market, agents in one side do it and agents in the other side stay in the market without making a decision (*inactive*). Note that the Patient algorithm requires the planner to know which agents will perish imminently if not matched. The information is referred to as the *departure information*.

Table 1: Summary of the loss when $T, \lambda, \lambda_a, \lambda_b \to \infty$. We denote $d = \lambda p$ and $d_i = \lambda_i p$ for $i \in \{a, b\}$, which are all constants.

(a) Lower bounds of the loss, and upper bounds of the loss when $\lambda_a = \lambda_b$

Setting		Loss	
		Lower bound	Upper bound ($d_a = d_b$)
non-bipartite	Greedy [2]	$\frac{1}{2d+1}$	$\frac{\log(2)}{d}$
	Patient [2]	$\frac{e^{-d}}{d+1}$	$\frac{e^{-d/2}}{2}$
2-sided	Greedy	$\max\left\{\Delta, \frac{1}{2d_a+d_b+1}\right\}$ (cf. [7])	$\frac{2\log(d_a+3)}{d_a}$
	Patient	$\frac{1}{2}\left(\frac{e^{-d_a}}{d_a+1} + \frac{e^{-d_b}}{d_b+1}\right)$	$e^{-O(d_a)}$
1-sided	Greedy	$\max\left\{\Delta, \frac{1}{1+2d_a+d_b}\right\}$	$\frac{2\log(d_a+3)}{d_a}$
	Patient	$\max\left\{\Delta, \frac{\log d_b}{d_a+d_b}\right\}$	

(b) Upper bounds when $\lambda_a \neq \lambda_b$. In the 2-sided market, we assume $\lambda_a \geq \lambda_b$, and in the 1-sided market, we assume that agents with rate λ_a are inactive.

Setting		Loss		
		Total	λ_a-side: $\mathbf{L}_a$	λ_b-side: $\mathbf{L}_b$
2-sided ($d_a \geq d_b$)	Greedy	$\Delta + \frac{2\log(d_b+3)}{d_a+d_b}$	$\frac{d_a-d_b}{d_a} + \frac{\log(d_b+3)}{d_a}$	$\frac{\log(d_b+3)}{d_b}$
	Patient	$\Delta + \frac{\log(d_b+3)}{d_a+d_b} + e^{-\max\left\{d_a-d_b, \frac{d_a}{1+d_b}\right\}}$		$e^{-\max\left\{d_a-d_b, \frac{d_a}{1+d_b}\right\}}$
1-sided ($d_a \geq d_b$)	Greedy	$\Delta + \frac{2\log(d_b+3)}{d_a+d_b}$	$\frac{\lvert d_a-d_b\rvert}{d_a} + \frac{\log(d_b+3)}{d_a}$	$\frac{\log(d_b+3)}{d_b}$
	Patient			
1-sided ($d_a < d_b$)	Greedy		$\frac{\log(d_b+3)}{d_a}$	$\frac{\lvert d_a-d_b\rvert}{d_b} + \frac{\log(d_b+3)}{d_b}$
	Patient			

Theoretical Guarantee Our main contributions are to derive theoretical bounds on the Greedy and Patient algorithms in the 2-sided and 1-sided markets, respectively. The obtained guarantees are summarized as in Tables 1(a) and 1(b). We here denote $d_i = \lambda_i p$ for $i \in \{a, b\}$ and $\Delta = \frac{|d_a-d_b|}{d_a+d_b}$. We remark that lower bounds for the 2-sided market model were also derived by Jiang [7].

Let us first consider the balanced case, that is, when $\lambda_a = \lambda_b$, implying that $d_a = d_b$ and $\Delta = 0$ in Table 1(a). Table 1(a) shows that the loss of the 2-sided Greedy algorithm is $\Theta\left(\frac{1}{d_a}\right)$, ignoring a logarithmic factor in d_a, while the 2-sided Patient algorithm has the loss $e^{-\Theta(d_a)}$. Thus waiting to match agents allows us to achieve exponentially small loss, which is a similar consequence to the non-bipartite matching market [2]. In contrast, the 1-sided market leads to a different conclusion. In fact, both of the Greedy and Patient algorithms have the same loss, which is $\Theta\left(\frac{1}{d_a}\right)$, ignoring a logarithmic factor in d_a. This means that waiting to match agents is not valuable in the 1-sided market, and other information such as the graph structure is necessary to achieve smaller loss.

The situation changes when $d_a \neq d_b$. For better understanding, we evaluate the proportion of unmatched agents on both sides separately, which are the losses $\mathbf{L}_a$ and $\mathbf{L}_b$ of λ_a-side and λ_b-side, respectively, in Table 1(b). Note that the total loss is equal to $\frac{d_a}{d_a+d_b}\mathbf{L}_a + \frac{d_b}{d_a+d_b}\mathbf{L}_b$. We see from Table 1(b) that the larger side, i.e., the side with $\max\{d_a, d_b\}$, has a constant loss of $\frac{|d_a-d_b|}{\max\{d_a,d_b\}}$ in every market. This factor is unavoidable since a bipartite graph is unbalanced. Our results say that, except for the unavoidable loss, we suffer only the loss of $O\left(\frac{1}{\max\{d_a,d_b\}}\right)$ on the large side in every market.

In the 2-sided market when $d_a \neq d_b$, the smaller side of the Patient algorithm has exponentially smaller loss than that of the Greedy algorithm. This again indicates that waiting to thicken the market in the 2-sided market is beneficial. In contrast, both of 1-sided Greedy and Patient algorithms have the same loss as the 2-sided Greedy algorithms.

We remark that, in the 1-sided Greedy algorithm, agents on one side do not attempt to match. Hence, it has less opportunity to make a partner compared to the 2-sided Greedy algorithm, which implies that the 1-sided Greedy algorithm seems to have larger loss. However, our results show that their losses have the same order. On the other hand, in the 1-sided Patient algorithm, since an active agent delays her decision, she is allowed to have more neighbors. Hence the 1-sided Patient algorithm intuitively has smaller loss than the 1-sided Greedy algorithm. However, our results show that their losses have the same order. In fact, Table 1(a) shows that the loss of the 1-sided Patient algorithm is strictly worse than the 2-sided one when $d_a = d_b$.

Another contribution of this paper is to evaluate the loss of optimal algorithms. We show that *any* algorithm suffers a loss of at least $1/(2d_a + d_b + 1)$ if it does not know the departure information. In other words, no matter how long each agent waits, the loss must be at least $1/(2d_a + d_b + 1)$. Thus the Greedy algorithm is almost optimal, up to a logarithmic factor in d_a. In contrast, if we know the departure information, we prove that the loss of *any* algorithm is at least $\frac{1}{2}\left(\frac{e^{-d_a}}{d_a+1} + \frac{e^{-d_b}}{d_b+1}\right)$. Thus, since the loss of the Patient algorithm is $e^{-O(d_a)}$ when $d_a = d_b$, waiting to match agents suffices to achieve optimal loss.

Technical Highlights The key observation for bounding the loss is that the number of agents in the market determines the loss of matching algorithms. This is observed in a non-bipartite market [2] as well. In our bipartite markets, in particular, the loss on one side is determined by the number of agents on the other side; an agent is likely to be matched if there are many agents on the other side, and the number of agents on the same side does not matter.

In the 2-sided market, since the Greedy algorithm attempts to match agents as soon as possible, the number of available agents on both sides is reduced rapidly when d_a and d_b grow (the market is *thin*). Since the market has no edges under the Greedy algorithm, all urgent agents perish, which are counted as the loss. On the other hand, the Patient algorithm attempts to match only urgent agents, which implies that the number of agents on both sides will remain large even when d_a and d_b increase (the market is *thick*). This allows the planner to find a pair to an urgent agent, which reduces the loss. We remark that, in the case when $d_a > d_b$, since the number of agents on the λ_b-side is small compared to the one on the λ_a-side, agents on the λ_a-side is hard to find a partner even if the market is thick, which worsens the loss of the larger side.

A similar observation can be applied to the 1-sided market. As observed in the 2-sided market, the market size of active agents (i.e., agents who can make a decision) will be thin under the Greedy algorithm, while it will be thick under the Patient algorithm. However, as we will see, the number of inactive agents decreases rapidly under both algorithms when d_a and d_b grow. This causes large loss for both the algorithms.

The above observation can be formalized with Markov chain. That is, the dynamics of our proposed algorithms can be modeled as continuous-time Markov chains determined by a pair of market sizes on both sides. We first show that the loss of the proposed algorithms can be expressed as the pool sizes in the stationary distribution of the Markov chain. Moreover, we prove that, for each of the proposed algorithms, the pool sizes in the stationary distribution highly concentrate around some values, which allows us to upper-bound the loss of the algorithms.

The most challenging part is to find the concentration of the pool sizes in the steady state. The primary technical tool is the balance equations of Markov chains. The balance equation describes the probability flux in and out of a given set of states. For a non-bipartite matching model [2], a Markov chain is of a simple form on the set of non-negative integers, and hence we can naturally apply the balance equations. On the other hand, our Markov chain is defined on 2-dimensional space, i.e., each state is a pair of market sizes. This requires us to choose a set of states for the balance equations more carefully. In fact, we need to adopt different strategies for each of the proposed algorithms.

References

[1] Atila Abdulkadiroglu and Tayfun Sönmez. Matching markets: Theory and practice. *Advances in Economics and Econometrics*, 1:3–47, 2013.

[2] Mohammad Akbarpour, Shengwu Li, and Shayan Oveis Gharan. Thickness and information in dynamic matching markets. *Journal of Political Economy*, 128(3):783–815, 2020.

[3] John Dickerson, Karthik Sankararaman, Aravind Srinivasan, and Pan Xu. Allocation problems in ride-sharing platforms: Online matching with offline reusable resources. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[4] Laura Doval. A theory of stability in dynamic matching markets. Technical report, Technical report, mimeo, 2014.

[5] Yuval Emek, Shay Kutten, and Roger Wattenhofer. Online matching: haste makes waste! In *Proceedings of the Forty-eighth Annual ACM Symposium on Theory of Computing*, pages 333–344, 2016.

[6] Monika Henzinger, Shahbaz Khan, Richard Paul, and Christian Schulz. Dynamic matching algorithms in practice. In *28th Annual European Symposium on Algorithms (ESA 2020)*, volume 173, page 58. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2020.

[7] Weiwei Jiang. Bipartite matching model with dynamic arrivals and departures. *International Journal of Modeling, Simulation, and Scientific Computing*, 9(04):1850031, 2018.

[8] Naonori Kakimura and Donghao Zhu. Dynamic bipartite matching market with arrivals and departures. *CoRR*, abs/2110.10824, 2021.

[9] Richard M. Karp, Umesh V. Vazirani, and Vijay V. Vazirani. An optimal algorithm for on-line bipartite matching. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pages 352–358, 1990.

[10] Morimitsu Kurino. Credibility, efficiency, and stability: A theory of dynamic matching markets. *The Japanese Economic Review*, 71(1):135–165, 2020.

[11] Aranyak Mehta. Online matching and ad allocation. *Foundations and Trends in Theoretical Computer Science*, 8(4):265–368, 2013.

[12] Luoyi Sun, Ruud H Teunter, Guowei Hua, and Tian Wu. Taxi-hailing platforms: Inform or assign drivers? *Transportation Research Part B: Methodological*, 142:197–212, 2020.

[13] Hai Wang and Hai Yang. Ridesourcing systems: A framework and review. *Transportation Research Part B: Methodological*, 129:122–155, 2019.

Composition Ordering for Linear Functions

Kazuhisa Makino[1]

Research Institute for Mathematical Sciences,
Kyoto University,
Kyoto, 606-8502, Japan
makino@kurims.kyoto-u.ac.jp

Abstract: We outline the composition ordering problem of linear functions, i.e., given n linear functions $f_1, \ldots, f_n : \mathbb{R} \to \mathbb{R}$ and a constant $c \in \mathbb{R}$, we construct a permutation $\sigma : [n] \to [n]$ that minimizes $f_{\sigma(n)} \circ f_{\sigma(n-1)} \circ \cdots \circ f_{\sigma(1)}(c)$, where $[n] = \{1, \ldots, n\}$. We discuss structual properties of optimal solutions for the problem as well as the current status of the complexity issue. We also consider the multiplication ordering of n matrices.

Keywords: Function composition, Matrix multiplication, Time-dependent scheduling, Flow shop Scheduling

Composition ordering for linear functions

The composition ordering problem of linear functions is given n linear functions $f_1, \ldots, f_n : \mathbb{R} \to \mathbb{R}$ and a constant $c \in \mathbb{R}$, to construct a permutation $\sigma : [n] \to [n]$ that minimizes (or maximizes) $f_{\sigma(n)} \circ f_{\sigma(n-1)} \circ \cdots \circ f_{\sigma(1)}(c)$, where $[n] = \{1, \ldots, n\}$.

For example, if the input consists of $f_1(x) = -\frac{1}{2}x + \frac{3}{2}$, $f_2(x) = x - 3$, $f_3(x) = 3x - 1$, and $c = 0$, then the permutation σ such that $\sigma(1) = 1$, $\sigma(2) = 2$ and $\sigma(3) = 3$ minimizes the objective value, while σ' such that $\sigma'(1) = 2$, $\sigma'(2) = 3$ and $\sigma'(3) = 1$ maximizes it. In fact, $f_3 \circ f_2 \circ f_1(0) = f_3(f_2(f_1(0))) = f_3(f_2(\frac{3}{2})) = f_3(-\frac{3}{2}) = -\frac{11}{2}$ is the optimal value of the minimization problem. Similarly, $f_1 \circ f_3 \circ f_2(0) = \frac{13}{2}$ is the optimal value of the maximization problem. The composition ordering problem is natural and fundamental in many fields such as artificial intelligence, computer science, and operations research. It is also known that the single machine*time-dependent scheduling* can be formulated as the composition ordering problem [3]. The problem is formulated as follows [1, 2]. Let J_i ($i \in [n]$) denote a job with a ready time $r_i \in \mathbb{R}$, a deadline $d_i \in \mathbb{R}$, and a processing time $p_i : \mathbb{R} \to \mathbb{R}$, where $r_i \leq d_i$ is assumed. Different from the classical setting, the processing time p_i is *not* a constant, but depends on the *starting* time of job J_i. The model has been studied to deal with learning and deteriorating effects, Here each p_i is assumed to satisfy $p_i(t) \leq s + p_i(t + s)$ for all t and $s \geq 0$, since we should be able to finish processing job J_i earlier if it starts earlier. Among time-dependent settings, we consider the single machine setting to minimize the makespan, where the input is the start time t_0 ($= 0$) and a set of J_i ($i \in [n]$) above. The makespan denotes the time when all the jobs have finished processing, and we assume that the machine can handle only one job at a time and preemption is not allowed.

We present an overview of the problem, which is based on [3] and a joint work with S. Kubo (Tottori University of Environmental Studies) and S. Sakamoto (Kyoto University). Especially, we provides several structual properties of optimal solutions for the problem, and show that it is computed in polynomial time if all functions are non-negative, and fixed-parameter tractable with respect to the number of negative functions. We also consider the generalization of the problem to the multiplication ordering of matrices.

[1]This work was partially supported by the joint project of Kyoto University and Toyota Motor Corporation, titled " Advanced Mathematical Science for Mobility Society " and JSPS KAKENHI Grant Numbers JP19K22841, JP20H00609, and JP20H05967.

References

[1] T.C.E. Cheng, Q. Ding, B.M.T. Lin, A concise survey of scheduling with time- dependent processing times, *EJOR* **152** (2004) 1-13.

[2] S. Gawiejnowicz, ATime-Dependent Scheduling, Springer 2008.

[3] Y. Kawase, K. Makino, and K. Seimi, Optimal composition ordering problems for piecewise linear functions, *Algorithmica* **80** (2018) 2134-2159.

Rigidity of Hypergraphs under Algebraic Constraints

Shin-ichi Tanigawa[1]

Department of Mathematical Informatics
University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo, Japan
tanigawa@mist.i.u-tokyo.ac.jp

Abstract: In this talk I will introduce a new rigidity notion for hypergraphs under algebraic constraints. This captures ordinary Euclidean rigidity, volume-constraint rigidity, the identifiability of partially-filled symmetric or skew-symmetric tensors, and more. I will explain new challenges emerged in the analysis of this new rigidity model.

This talk is based on a joint work with James Cruickshank, Fatemeh Mohammadi, and Anthony Nixon.

Keywords: graph rigidity, matroids, hypergraphs, tensors

1 Introduction

Suppose there are n points $p_1, p_2, \ldots, p_n$ in $\mathbb{R}^d$ whose positions are unknown, and suppose that one can get relations among those points though a measuring device. The question we want to address is the identifiability of the points from measurements.

In this work we formulate this problem as follows. Let g be a polynomial map, which is a measurement function representing a measurement device, and suppose that the value of g is determined for each tuple of k points in $\mathbb{R}^d$. Assuming that g is a symmetric k-form over $\mathbb{R}^d$, the set of observations an observer can get is represented by a k-uniform hypergraph G. Namely the observe can get $g(p_{i_1}, \ldots, p_{i_k})$ for all $\{i_1, \ldots, i_k\} \in E(G)$. Then the identifiability problem asks if the polynomial system

$$g(x_{i_1}, \ldots, x_{i_k}) = g(p_{i_1}, \ldots, p_{i_k}) \qquad (\{i_1, \ldots, i_k\} \in E(G))$$

in variables $x_1, x_2, \ldots, x_n \in \mathbb{R}^d$ has a unique solution (up to certain symmetry). This formulation is based on graph rigidity theory, which addresses the case when g is the Euclidean distance between two points.

Since the idea of graph rigidity is so natural, various variants of the graph rigidity problem has been already introduced. Typical such examples are rigidity in different metric spaces such as spherical space or L_p-space and rigidity with respect to other geometric constraints. It is possible to consider further general algebraic systems, but we might loose the combinatorial perspective, which is the core of rigidity theory. In this talk, we see that the new rigidity model of hypergraphs is general enough to capture the identifiability problem in several applications and it also gives new mathematical challenges.

2 Rigidity of Hypergraphs: Formal Definition

Let $\binom{X}{k}$ be the set of multisets of k elements of a finite set X and $\binom{X}{k}$ be the subset of $\binom{X}{k}$ consisting of sets having no repeated elements. Throughout the paper, a *k-uniform hypergraph* G is defined as a pair $G = (V, E)$ of a finite set and $E \subseteq \binom{V}{k}$, and G is said to be *simple* if $E \subseteq \binom{V}{k}$.

[1]Research is supported by JST PRESTO Grant Number JPMJPR2126 and JSPS KAKENHI Grant Number 18K11155.

Let $\mathbb{F}$ be either $\mathbb{R}$ or $\mathbb{C}$. Extending the central object from rigidity, a pair (G, p) of a hypergraph G and $p : V \to \mathbb{F}^d$ is called a *d-dimensional hyper-framework* or a *hyper-framework in $\mathbb{F}^d$*.

Suppose we are given a k-uniform hyper-framework (G, p) and $g : (\mathbb{F}^d)^k \to \mathbb{F}$ be a k-form over $\mathbb{F}^d$. For each hyperedge $e = \{v_1, \ldots, v_k\} \in E$, define $f_e : (\mathbb{F}^d)^V \to \mathbb{F}$ by $f_e(p) = g(p(v_1), \ldots, p(v_k))$. The tuple $f_{g,G} := (f_e : e \in E)$ is regarded as a polynomial map $f_{g,G} : (\mathbb{F}^d)^V \to \mathbb{F}^E$, which is called the *$g$-measurement map* of G.

Note that, in order to make f_e (and hence $f_{g,G}$) well-defined, g must be either *symmetric* or *anti-symmetric* k-form (i.e., symmetric or anti-symmetric with respect to the ordering of points), and if g is anti-symmetric we always suppose that $v_1, \ldots, v_k$ are aligned in increasing order in $f_e(p) = g(p(v_1), \ldots, p(v_k))$, assuming a (fixed) total order on the vertex set of G.

In most practical applications, there is a nontrivial group action that does not change the value of a g-measurement map, and the rigidity has to be defined by taking care of the degree of freedom caused by such actions. Let Γ be a subgroup of the general affine group $\mathrm{Aff}(d, \mathbb{F})$, which consists of pairs (A, t) of $A \in \mathrm{GL}(d, \mathbb{F})$ and $t \in \mathbb{F}^d$. The action of Γ to $(\mathbb{F}^d)^V$ is defined by $(\gamma \cdot p)(v) = Ap(v) + t$ for any $\gamma = (A, t) \in \Gamma$ and $p \in (\mathbb{F}^d)^V$. We say that $\gamma = (A, t)$ *stabilizes* $g : (\mathbb{F}^d)^k \to \mathbb{F}$ if $g(\gamma \cdot q) = g(q)$ for any $q \in (\mathbb{F}^d)^k$. The set of pairs (A, t) that stabilize g forms a subgroup of $\mathrm{Aff}(d, \mathbb{F})$, called the *stabilizer* of g.

Suppose that Γ is the stabilizer of g. Then Γ is also the stabilizer of f. We say that (G, p) is *globally g-rigid* if for any $q \in f_{g,G}^{-1}(f_{g,G}(p))$ there is $\gamma \in \Gamma$ such that $q = \gamma \cdot p$. We say that (G, p) is *locally g-rigid* if there is an open neighbour N of p in $(\mathbb{F}^d)^V$ such that for any $q \in f_{g,G}^{-1}(f_{g,G}(p)) \cap N$ there is $\gamma \in \Gamma$ such that $q = \gamma \cdot p$.

3 Examples

We give a list of primary examples.

Ordinary Euclidean rigidity. A fundamental example is the case when G is 2-uniform (i.e., a graph) and $g(x, y) = \sum_{i=1}^{d}(x_i - y_i)^2$ for $x, y \in \mathbb{R}^d$. Then the Euclidean group $E(d)$ is the stabilizer of g, and its Lie algebra is the set of pairs (S, t) of skew-symmetric matrices S and $t \in \mathbb{R}^d$. In this case, g-rigidity is nothing but the standard rigidity of frameworks.

Rigidity in pseudo-Euclidean space. A closely related example comes from changing the underlying metric of the space to a pseudo-Euclidean metric such as in Minkowski space. In this context G is 2-uniform (i.e., a graph) and $g(x, y) = \sum_{i=1}^{d_1}(x_i - y_i)^2 - \sum_{i=d_1+1}^{d}(x_i - y_i)^2$ for $x, y \in \mathbb{R}^d$ and $d_1 \leq d$.

L_p-norm rigidity. An alternative generalization is to allow the distance function to be replaced by distance under another norm. Specifically G is 2-uniform (i.e., a graph) and $g(x, y) = \sum_{i=1}^{d} |x_i - y_i|^p$ for $x, y \in \mathbb{R}^d$ and $1 < p < \infty$.

Volume-constrained rigidity. Given a d-dimensional pure simplicial complex realized in $\mathbb{R}^d$, the notion of volume-constrained rigidity concerns whether there is a motion of vertices keeping the (signed) volume of each $(d-1)$-simplex. A d-dimensional pure simplicial complex can be identified with a $(d+1)$-uniform hyper-framework (G, p) with a simple hypergraph G. Hence, the volume-constrained rigidity can be captured as the g-rigidity of a simple $(d+1)$-uniform hyper-framework (G, p) with $g : (\mathbb{R}^d)^{d+1} \to \mathbb{R}$ defined by

$$g(p_1, p_2, \ldots, p_{d+1}) = \det \begin{pmatrix} p_1 & p_2 & \cdots & p_{d+1} \\ 1 & 1 & \ldots & 1 \end{pmatrix}.$$

Positive semidefinite symmetric matrix completion. Let T be a positive semidefinite symmetric matrix of size n over $\mathbb{R}$. If the rank of T is r, then the spectral decomposition implies that

$$T = \sum_{i=1}^{r} x_i^\top x_i \tag{1}$$

for some vectors $x_1, x_2, \ldots, x_r \in \mathbb{R}^n$. Let P be an $r \times n$-matrix obtained by aligning x_i as the i-th row vector, and define $p : [n] \to \mathbb{R}^r$ such that $p(i)$ is the i-th column of P. Then $p(i) \cdot p(j)$ is equal to the (i, j)-th entry t_{ij} of T.

In the positive semidefinite symmetric matrix completion problem, we are given a partially-filled positive semidefinite symmetric matrix T and asked to recover the positive semidefinite symmetric matrix by filling missing entries. If we use a graph $G = ([n], E)$ (which may contain self-loops) to denote a set of indices $e = (i, j)$ of given entries t_e of T, then the problem is to find $p : [n] \to \mathbb{R}^r$ satisfying

$$p(i) \cdot p(j) = t_e \qquad (e = \{i, j\} \in E). \tag{2}$$

We are, in particular, interested in the uniqueness of the completions rather than finding a completion. In the unique completion problem, we are given a solution $p : [n] \to \mathbb{R}^r$ of (2) and are asked if it is the unique solution of (2). This is a property of a framework (G, p) and is captured by the g-rigidity.

Specifically, consider a 2-uniform hypergraph (i.e., a graph) G and $g(x, y) = \sum_{i=1}^{d} x_i y_i$ for $x, y \in \mathbb{R}^d$. Then the orthogonal group $O(d)$ is the stabilizer of g, and the global g-rigidity decides if (2) has the unique solution up to the action of $O(d)$, or equivalently a completion is unique.

Symmetric tensor completions. The idea in the last paragraph can be extended to tensors as follows.

For a vector space V over $\mathbb{C}$, let $V^{\otimes k}$ be the k-fold tensor product of V. We fix a basis of V, and assume that each $T \in V^{\otimes k}$ is represented by a k-dimensional array of numbers in $\mathbb{C}$. A tensor $T \in V^{\otimes k}$ is said to be *symmetric* if for any permutation σ on $[k]$ we have $T_{i_1, i_2, \ldots, i_k} = T_{\sigma(i_1), \sigma(i_2), \ldots, \sigma(i_k)}$. The set of symmetric tensors in $V^{\otimes k}$ is denoted by $S^k(V)$. It is known that any symmetric tensor can be written as

$$T = \sum_{i=1}^{r} x_i^{\otimes k} := \sum_{i=1}^{r} x_i \otimes x_i \otimes \cdots \otimes x_i \tag{3}$$

for some vectors $x_1, x_2, \ldots, x_r \in V$.

For $T \in S^k(V)$, the smallest possible r for which we can write T in the form of (3) is called the *symmetric rank* of T. The subset of $S^k(V)$ consisting of tensors with symmetric rank at most r is denoted by $S^k_r(V)$. When $V = \mathbb{C}^n$, each element in $S^k(V)$ is called a *symmetric tensor of order k of size n* (over $\mathbb{C}$).

Once we introduce a notion of rank, the corresponding low-rank completion problem can be defined automatically. In the symmetric tensor completion problem, given a partially-filled tensor of order k and size n, we are asked to fill the remaining entries to obtain a symmetric tensor of symmetric rank at most r. Recall that $\left\{{X \atop k}\right\}$ denotes the set of multisets of k elements of a finite set X. Due to the symmetry condition, each entry of a symmetric tensor can be indexed by an element in $\left\{{X \atop k}\right\}$. Hence, we can use a subset E of $\left\{{[n] \atop k}\right\}$ to represent the known entries in the symmetric tensor completion problem. In this manner, we encode the underlying combinatorics of each instance of the completion problem using a k-uniform hypergraph $([n], E)$, and the symmetric tensor completion problem can be reformulated as a hypergraph embedding problem as follows: Given a k-uniform hypergraph $G = ([n], E)$ and $a_e \in \mathbb{C}$ for $e \in E$, find $p : [n] \to \mathbb{C}^r$ such that

$$\mathbf{1} \cdot \bigodot_{v \in e} p(v) = a_e \qquad \text{for } e \in E, \tag{4}$$

where $\mathbf{1}$ denotes the all-one vector and $\bigodot$ denotes the Hadamard product of vectors, that is the component-wise product.

The corresponding unique completion problem can be captured by g-rigidity for $g : (\mathbb{C}^r)^k \to \mathbb{C}$ defined by

$$g(p_1, \ldots, p_k) = \mathbf{1} \cdot \bigodot_{i \in \{1,\ldots,k\}} p_i.$$

The stabilizer Γ_g is the set of diagonal matrices over $\mathbb{C}$ whose diagonal entries are k-th roots of unity.

Sorting columns of a matrix to optimize nondecreasing subsequences of rows

Naoki Fujihara, Takeshi Tokuyama[1]

Department of Computer Science,
Kwansei Gakuen University, Japan
tokuyama@kwansei.ac.jp

Abstract: Given a set of n vectors with d dimensions, we consider a $d \times n$ matrix arranging them as column vectors. The matrix depends on the order of the column vectors, and we consider the problems of finding the optimal permutation of column vectors to maximize/minimize an objective function. Our objective function is defined by using non-decreasing subsequences of row vectors of the matrix. A non-negative utility is given to each element of the matrix. First, we consider the the first-fit nondecreasing subsequence of each row, and find the permutations of the columns such that the total utility of elements in the first-fit non-decreasing subsequences of rows is maximized and minimized. We investigate the complexity of the problems, and give polynomial time algorithms if d is a constant. We also consider several other models to formulate the column-vector arrangement problem and their solutions.

1 Introduction

1.1 Column arrangement problems of matrices

Arranging a data set in a suitable order is a common operation in computer science[4, 6, 8, 10]. If the data set has a total order (e.g., a set of real numbers), it is natural to arrange the set in either non-decreasing or non-increasing order. The operation is the *sorting*. However, it is nontrivial to define a suitable order for a general data set, and often difficult to compute even if it is defined. For example, Hamiltonian path problem is to find an ordered list of all the vertices of a graph so that each adjacent vertices in the list are connected by an edge of the graph.

In particular, it is an important problem to arrange a set of d-dimensional vectors (say, real vectors) in a suitable way, which is called *multi-dimensional sorting*. There is no universal way of multi-dimensional sorting, and it depends on applications to organize the data (e.g., data structures for multi-dimensional searching [11]). In this paper, we give a model of multi-dimensional sorting as arranging column vectors of a matrix using non-decreasing subsequences in its rows.

Given a sequence $\mathbf{a} = (a_1, a_2, \ldots, a_n)$ of elements in a totally ordered set (say, integers or real numbers), its *first-fit non-decreasing subsequence* is the subsequence obtained by reading the sequence from left (i.e., in the increasing order of the index) and selecting a_i if and only if it is not less than any elements read before. For example, if $\mathbf{a} = (3, 1, 3, 5, 2, 4, 6)$, its first-fit non-decreasing subsequence is $(3, 3, 5, 6)$. In other words, it is the maximal non-decreasing sequence starting from a_1 generated by the greedy algorithm appending the next possible element to extend subsequence. We consider a nonnegative-valued function φ called *utility function* on the elements of $\mathbf{a}$, and the utility $\Phi(\mathbf{a})$ of $\mathbf{a}$ is defined as the sum of utility function values of entries of the first-fit non-decreasing subsequence of $\mathbf{a}$. In the above example, $\Phi(\mathbf{a}) = \varphi(3) + \varphi(3) + \varphi(5) + \varphi(6)$.

Consider a $d \times n$ matrix $\mathbf{A}$ of elements so that a total order is given for entries of each row. Its utility $\Phi(\mathbf{A})$ is the sum of the utilities of all rows of $\mathbf{A}$. We apply a permutation σ of columns to obtain a matrix $\sigma\mathbf{A}$ to optimize the utility. We consider the Utility Maximizing Column Arrangement (**MaxCA**) and Utility Minimizing Column Arrangement (**MinCA**) problems to find the permutations σ maximizing and minimizing $\Phi(\sigma\mathbf{A})$, respectively.

[1]Supported by MEXT JSPS KAKENHI Grant-in-Aid for Scientific Research(B) 20H04143.

1.2 Motivations

Sorting and maximum element finding:

If $d = 1$, $\mathbf{A}$ is a sequence $\mathbf{a}$. For any positive-valued utility function, the utility is maximized if and only if the sequence $\sigma\mathbf{a}$ is a non-decreasing sequence. Thus, **MaxCA** for $d = 1$ is equivalent to the sorting problem, and hence solved in $O(n \log n)$ time. The utility is minimized if and only if the sequence $\sigma\mathbf{a}$ satisfies that its first entry is the maximum element. Thus, **MinCA** for $d = 1$ is equivalent to the maximum element finding, and hence solved in $O(n)$ time. Therefore, the column arrangement problems can be considered as generalizations of sorting and maximum element finding to a set of points in d-dimensional space if the entries of $\mathbf{A}$ are real numbers. Especially, if $\varphi \equiv 1$, **MaxCA** is to maximize the sum of lengths of the first-fit non-decreasing sequences in the rows.

Witch and brave in fantasy lands:

MaxCA was inspired by a task that occurred in the plot of a *crafting game* that the first author (a master course graduate student who sought for a job in a farm developing video games) proposed. Imagine that a witch prepares a magic potion. She blends several materials: herbs, spices, magic carrot, dried frog, etc. The order to add the materials in the pot is important: If x is added wrongly before y that should be added earlier than x, the utility of y disappears. Now, consider a more complicated situation that each material contains constituents in two or more categories (say, *hypnotic-factor* and *aphrodisiac-factor* categories), and the order is given for each category separately. The orderings may be incompatible: For example, imagine that the magic carrot should be added earlier than the dried frog to activate its utility of hypnotic factor, but it should be added later to activate the utility of aphrodisiac factor of the dried frog. Then, what is the best order to add materials to maximize the total utility? This question is formulated as **MaxCA**.

We may also consider the problems as variations of the prize-collecting traveling salesman problem [2] as shown in the following story. In Japanese role-playing video games, such as *Dragon Quest* (named *Dragon Warrior* in US) and *Final Fantasy*, the main character (the brave) travels and grows to a legendary hero through experience. The brave visits shops to buy equipment items in d categories, e.g. wears (including armors), tools, accessories, and weapons. The set of items sold depends on the shops. We assume that items in a category are totally ordered by the levels of their power. For example, a knife is a weak weapon, and the bronze sword is better, but the magic sword is far better. The brave can simultaneously mount at most one item of each category. So, the brave is eager to buy if a shop sells a better item than he/she has. The brave cannot foresee the future, so his/her action is greedy.

The order to visit shops is usually guided by the scenario of games. However, one may wonder what is the best order to visit shops. If there are n shops, it makes a $d \times n$ matrix $\mathbf{A}$ such that each column corresponds to the set of items in a shop. One candidate objective is the total amount of money to spend. Then, a seemingly good (although not always the best) strategy is to buy the best equipment items as early as possible to avoid spending money for less-valuable ones. The problem can be formulated as **MinCA**, where $\varphi(x)$ is the price of x. However, in the role-playing games, the brave needs to fight against monsters to earn money to buy expensive goods, and it is required to be armed with some inexpensive equipment in early stages. Thus, in order to proceed the game efficiently, generally it is better to buy many equipment items gradually from weaker to stronger. Therefore, the problem is formulated as **MaxCA** where $\varphi(x)$ corresponds to the profit given by buying the equipment x. **MaxCA** can give the maximum profit ordering for the owner of shops, too.

1.3 Related previous results

Although the authors do not know previous works on exactly the same problem, there are works on related problems. As shown in Section 4.1, **MinCA** for a binary matrix is equivalent to the min-sum set covering problem (**MSSC**) proposed by Feige-Lovász-Tetali [7]. The min-sum set covering problem is known to be NP-hard. On the positive side, a greedy algorithm to give a 4 approximation solution is known, and the approximation factor is almost tight.

The problem of arranging data to optimize a given objective function is called *linear ordering (or arrangement) problem* in the literature. A famous example is the linear arrangement problem of graphs[6, 8], where a bijection f from the vertex set V of a graph $G = (V, E)$ to $\{1, 2, \ldots, n\}$ minimizing (or maximizing) the summation $\sum_{(u,v)\in E} |f(u) - f(v)|$ is computed. A unified framework including both the min-sum set covering problem and the linear arrangement problem on graphs was given by Iwata *et al.*[9] named Minimum Linear Ordering Problem (**MLOP**) in which the objective function is summation of the values of submodular/supermodular functions defined by using prefixes (or suffixes) of the sequence. The problem is NP-hard in general, and approximation algorithms were studied. **MaxCA** for the binary case can be converted to a **MLOP** for a monotone submodular function.

1.4 Results

The results of this paper include the following:
1. If d is a constant, **MaxCA** and **MinCA** are computable in $O(n^{d+1})$ time.
2. Both **MinCA** and **MinCA** are NP-hard if d is a parameter of the time complexity even if the matrix **A** is binary.
3. Variations using the maximum non-decreasing subsequences (instead of the first-fit non-decreasing subsequence) are proposed.
4. Variations using cumulative utility functions as the objective functions are considered, for which approximation algorithms are given.

2 The MaxCA and MinCA Problems

2.1 The problem formulation

A totally ordered set X is a set with an ordering $<$ such that for any $x \neq y \in X$ either $x < y$ or $y > x$ holds. Therefore, for a finite set S of elements in X, its maximum element $\max\{x \in S\}$ with respect to $<$ always exists. The set $\mathbb{Z}_{\geq 0}$ of nonnegative integers is a typical example, and readers can imagine the totally ordered sets in the following arguments are $\mathbb{Z}_{\geq 0}$ to get intuition.

Given a sequence $\mathbf{a} = (a_1, a_2, \ldots, a_n)$ of elements in a totally ordered set, its i-th prefix-maximum is defined by $\mathrm{Max}(\mathbf{a}, \leq i) = \max_{j\leq i}\ a_j$.

An entry a_i of $\mathbf{a}$ is called a *prefix-max element* if $a_i = \mathrm{Max}(\mathbf{a}, \leq i)$. In other words, a_i is not smaller than any of $a_1, a_2, \ldots, a_{i-1}$. Let $\mathcal{M}(\mathbf{a})$ be the subsequence consisting of all prefix-max elements of $\mathbf{a}$. In other words, it is the maximal non-decreasing sequence starting from a_1 generated by the greedy algorithm, which we call the *first-fit* algorithm, appending the first possible element to extend the non-decreasing subsequence. Thus, we call $\mathcal{M}(\mathbf{a})$ the *first-fit maximal non-decreasing subsequence* of $\mathbf{a}$.

We consider a nonnegative-valued function φ called *utility* on the set of entries of $\mathbf{a}$, and let $\Phi(\mathbf{a}) = \sum_{a_i\in\mathcal{M}(\mathbf{a})} \varphi(a_i)$.

Consider a $d \times n$ matrix $\mathbf{A}$ so that total ordering is given for elements of each row. We assume a utility function φ is defined on the set of elements of $\mathbf{A}$. Let $\mathbf{A}(i)$ be the i-th row vector of $\mathbf{A}$. We call $\Phi(\mathbf{A}) = \sum_{1\leq i\leq d} \Phi(\mathbf{A}(i))$ the *utility* of $\mathbf{A}$.

Let $\mathcal{S}_n$ be the set of all permutations of $\{1, 2, \ldots, n\}$. Given $\sigma \in \mathcal{S}_n$, $\sigma\mathbf{A}$ is the matrix obtained by permuting the column indices of $\mathbf{A}$ by σ; that is, the (i, j) entry of $\sigma\mathbf{A}$ is $a_{i,\sigma(j)}$. We consider the following two problems:

1. **MaxCA** : Find $\sigma \in \mathcal{S}_n$ maximizing $\sum_{1\leq i\leq d} \Phi(\sigma\mathbf{A}(i))$.
2. **MinCA** : Find $\sigma \in \mathcal{S}_n$ minimizing $\sum_{1\leq i\leq d} \Phi(\sigma\mathbf{A}(i))$.

Example 1 *Consider a* 2×6 *matrix* $\mathbf{A} = \begin{pmatrix} \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{4} & \mathbf{5} & \mathbf{6} \\ \mathbf{6} & 2 & 3 & 5 & 4 & 1 \end{pmatrix}$. *For the uniform utility function* $\varphi(x) = 1$ *for all* x*, we can observe that* $\Phi(\mathbf{A}) = 6 + 1 = 7$*. Here, the entries written in red boldface show*

the first-fit non-decreasing subsequence of each row.

If $\sigma = (2,3,4,5,1,6)$, $\sigma\mathbf{A} = \begin{pmatrix} \mathbf{2} & \mathbf{3} & \mathbf{4} & \mathbf{5} & 1 & \mathbf{6} \\ \mathbf{2} & \mathbf{3} & \mathbf{5} & 4 & \mathbf{6} & 1 \end{pmatrix}$ *and* $\Phi(\sigma\mathbf{A}) = 5+4 = 9$. *On the other hand, for* $\tau = (1,6,2,3,4,5)$, $\tau\mathbf{A} = \begin{pmatrix} \mathbf{1} & \mathbf{6} & 2 & 3 & 4 & 5 \\ \mathbf{6} & 1 & 2 & 3 & 5 & 4 \end{pmatrix}$ *and* $\Phi(\tau\mathbf{A}) = 2+1 = 3$.

If we replace the utility function by the identity function $\varphi(x) = x$, *then* $\Phi(\mathbf{A}) = 21+6 = 28$, $\Phi(\sigma\mathbf{A}) = 20+16 = 36$, *and* $\Phi(\tau\mathbf{A}) = 7+6 = 13$.

In this example, for both utility functions, σ *and* τ *are solutions of* **MaxCA** *and* **MinCA**, *respectively. However, the solutions depend on* φ *in general.*

2.2 Difference between *non-decreasing* and *increasing* subsequences

We consider non-decreasing subsequences. If each row of the matrix is multiplicity-free (i.e., no same element occurs more than once), 'non-decreasing' and 'increasing' are the same. However, the difference matters in general.

In order to demonstrate the difference, let us consider the special case of **MinCA** where the utility function is uniform, that is, $\varphi(x) = 1$ for any x. In other words, we consider the maximum length greedy non decreasing subsequence of each row, and minimize the total length of them.

For simplicity, we assume $d \le n$. Since **MinCA** is a generalization of the maximum-element-finding problem, we may naturally consider the following greedy algorithm:

MMF (Move Maximum elements to the Front):
do while there is a remaining row;
1. Pick a suitable row and find the column containing the maximum element in the row;
2. Append the column to the output sequence of the column vectors;
3. Remove the row and the column;

end while;
4. Append remaining $n-d$ columns to the output sequence in any order;

We can observe that the total utility of the **MMF** algorithm is at most $d(d+1)/2$. On the other hand, if we consider the $d \times (d+1)$ matrix $\mathbf{A}$ that has elementary vectors $\mathbf{e}_i$ $(i = 1,2,\dots d)$ and the vector $\mathbf{v} = \frac{1}{2}(1,1,\dots 1)^\top$ as its column vector, the algorithm indeed has the total utility $d(d+1)/2$ for any order of selection of the row in step 1, while the optimal solution has $\mathbf{v}$ as the first column and attains the total utility $2d$. So, the approximation ratio of **MMF** is at least $(d+1)/4$, which is depressing.

However, if we consider the increasing sequences instead of non-decreasing sequences, we can improve the approximation ratio by slightly modifying the algorithm so that the step 1 selects the row randomly. Then, the expectation of the total utility is bounded by $d\ln d + o(d\ln d)$. Since the total utility is at least d, the randomized **MMF** algorithm attains a $\ln d + o(\ln d)$ approximation ratio. Accordingly, the difficulty of the problem may depend on existence of duplicate elements in the non-decreasing subsequences.

3 Algorithms for low dimensional cases

3.1 Graph representation of column arrangements

If d is a constant, we can design dynamic programming algorithms for each of maximization and minimization problems, which we explain as maximum and minimum weight path problems on a graph. For convenience' sake, we assume that totally ordered set for each row has an element that is smaller than every entry of the row, and it is denoted by a shared symbol $*$.

For two d-dimensional vectors $\mathbf{u}$ and $\mathbf{v}$, we say $\mathbf{u} \ge \mathbf{v}$ if $u_i \ge v_i$ for $i = 1,2,\dots d$. We say $\mathbf{u} > \mathbf{v}$ if $\mathbf{u} \ge \mathbf{v}$ and $\mathbf{u} \ne \mathbf{v}$. Given two d-dimensional vectors $\mathbf{u}$ and $\mathbf{v}$, their *max-join* $\mathbf{u} \oplus \mathbf{v}$ is their entry-wise maximum. That is, the i-th entry of $\mathbf{u} \oplus \mathbf{v}$ is $\max\{u_i, v_i\}$. By definition, $\mathbf{u} \oplus \mathbf{v} \ge \mathbf{u}$ and $\mathbf{u} \oplus \mathbf{v} \ge \mathbf{v}$.

Given a submatrix $\mathbf{B}$ consisting of column vectors of $\mathbf{A}$, its *signature* $\Lambda(\mathbf{B})$ is a d-dimensional vector whose i-th entry is the maximum element of the i-th row of $\mathbf{B}$. In other words, $\Lambda(\mathbf{B})$ is the max-join of all column vectors of $\mathbf{B}$. We artificially define $s = \Lambda(\emptyset) = (*, *, \ldots, *)$ for the empty submatrix $\emptyset$.

Let V_i be the set of elements of the i-th row of $\mathbf{A}$, and let $V' = V_1 \times V_2 \times, \ldots, \times V_d$ be their direct product. We define $V = V' \cup \{s\}$. Clearly, any signature $\Lambda(B)$ is in V, and hence there are at most $n^d + 1$ different signatures.

We define a weighted directed graph $G(\mathbf{A}) = (V, E)$ as follows: For each column vector $\mathbf{c}$ and each vector $\mathbf{v} \in V$, we define a directed edge $e \in E$ from $\mathbf{v}$ to $\mathbf{v} \oplus \mathbf{c}$ if $\mathbf{v} \oplus \mathbf{c} \neq \mathbf{v}$. Suppose that an edge e is from $\mathbf{v}$ towards $\mathbf{v}'$. By definition, $\mathbf{v}' > \mathbf{v}$. If $v'_i > v_i$ for an $i \in \{1, 2, \ldots, d\}$, we say v'_i is a *renewed entry* with respect to e. We define the weight $w(e)$ by the summation of the utilities of renewed entries with respect to e. Since each edge is corresponding to a column vector, a path in $G(\mathbf{A})$ corresponds to a sequence of column vectors of $\mathbf{A}$.

Example 2 *Let* $\mathbf{A} = \begin{pmatrix} \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{4} & \mathbf{5} & \mathbf{6} \\ \mathbf{6} & 2 & 3 & 5 & 4 & 1 \end{pmatrix}$ *and* $\varphi(x) = x$. *If* $\mathbf{v} = (3, 5)^\top$ *(*$\mathbf{x}^\top$ *is the transpose of* $\mathbf{x}$*) and* $\mathbf{c} = (5, 4)^\top$, *there is an edge* e *from* $\mathbf{v}$ *to* $\mathbf{v} \oplus \mathbf{c} = (\mathbf{5}, 5)^\top$. *The renewed entry is* 5 *shown in green. Thus, the weight is* $\varphi(5) = 5$.

Example 3 *The permutation* $\sigma = (2, 3, 4, 5, 1, 6)$ *gives* $\sigma\mathbf{A} = \begin{pmatrix} \mathbf{2} & \mathbf{3} & \mathbf{4} & \mathbf{5} & 1 & \mathbf{6} \\ \mathbf{2} & \mathbf{3} & \mathbf{5} & 4 & \mathbf{6} & 1 \end{pmatrix}$. *It corresponds to the path* $s = (*, *)^\top, \mathbf{v}_1 = s \oplus (2, 2)^\top = (\mathbf{2}, \mathbf{2})^\top, \mathbf{v}_2 = \mathbf{v}_1 \oplus (3, 3)^\top = (\mathbf{3}, \mathbf{3})^\top, \mathbf{v}_3 = \mathbf{v}_2 \oplus (4, 5)^\top = (\mathbf{4}, \mathbf{5})^\top, \mathbf{v}_4 = \mathbf{v}_3 \oplus (5, 4)^\top = (\mathbf{5}, 5)^\top, \mathbf{v}_5 = \mathbf{v}_4 \oplus (1, 6)^\top = (5, \mathbf{6})^\top, \mathbf{v}_6 = \mathbf{v}_5 \oplus (6, 1)^\top = (\mathbf{6}, 6)^\top = \Lambda(\mathbf{A})$. *The sequence of weights of edges along the path is* $4, 6, 9, 5, 6, 6$, *and the total weight is* 36, *which equals* $\Phi(\sigma\mathbf{A})$ *(See Example 1).*

Lemma 1 $G(\mathbf{A})$ *is a directed acyclic graph with* $O(n^d)$ *vertices and* $O(n^{d+1})$ *edges, and the sequence of edges in a directed pass from* s *to a vertex* $\mathbf{v}$ *corresponds to a sequence of columns of* $\mathbf{A}$ *without repetition.*

Accordingly, $\mathbf{v} \in V$ that is reachable from s in $G(\mathbf{A})$ is a signature of some matrix consisting of a set of columns of $\mathbf{A}$.

3.2 Algorithms for the multiplicity-free case

First, we consider the *multiplicity-free* case, where each element of V_i appears in the i-th row only once.

Theorem 1 *For the multiplicity-free case, the utilities of the optimal solution for* **MaxCA** *(resp.* **MinCA***) equals the weight of the maximum (resp. minimum) weight path from* s *to* $t = \Lambda(\mathbf{A})$ *of* $G(\mathbf{A})$. *Consequently,* **MaxCA** *and* **MinCA** *can be solved in* $O(n^{d+1})$ *time, respectively.*

Proof: Lemma 1 implies that a path corresponds to a sequence of columns, and it is routine to examine that the weight of a path equals the utility of the corresponding column sequence. We can compute the maximum weight path and minimum weight path in a directed acyclic graph in linear time in the number of edges by first topologically sort the vertices and then proceed a dynamic programming in the topological order.Hence, the time complexity is $O(n^{d+1})$. The permutation σ is obtained from the sequences of columns corresponding to the paths. The length of the obtained path may be shorter than n, and we append the rest of indices in an arbitrary manner to obtain σ. □

3.3 Algorithms for matrices with repeated entries

If there are repeated entries in a row of the matrix, we need careful treatment to solve **MaxCA**. The following example explains the difficulty.

Example 4 *Suppose that* $\mathbf{c} = (3, 1, 1)$ *is a column of* $\mathbf{A}$*, and a vertex* $\mathbf{v} = (3, a, b) = \Lambda(\mathbf{B})$ $(a > 1, b > 1)$ *corresponds to a submatrix* $\mathbf{B}$ *of* $\mathbf{A}$ *that does not contain the column* $\mathbf{c}$*. If we append* $\mathbf{c}$ *at the end of* $\mathbf{B}$*, we add the utility* $\varphi(3)$*, since the first entry* 3 *of* $\mathbf{c}$ *is appended to the maximal non-decreasing sequence in the first row. However,* $\mathbf{v} \oplus \mathbf{c} = \mathbf{v}$*, and hence there is no corresponding edge in our graph* $G(\mathbf{A})$*. If we add that edge, it makes a loop edge from* $\mathbf{v}$ *to* $\mathbf{v}$*. This destroys acyclicity and ruins the algorithm.*

Thus, instead of adding edges to create loops, we modify the weight of edges to apply a *deferred evaluation* strategy.

Suppose that an element x_i appears in the i-th row of $\mathbf{A}$ multiple times. For any vertex $\mathbf{v}$, $\mathbf{v}(\bar{i})$ is the $d-1$ dimensional vector obtained by removing the i-th entries of $\mathbf{v}$. Then, the multiplicity $m(x_i; \mathbf{v}(\bar{i}))$ of x_i bounded by $\mathbf{v}(\bar{i})$ is the number of columns $\mathbf{c}$ of $\mathbf{A}$ such that $c_i = x_i$ and $\mathbf{v}(\bar{i}) \geq \mathbf{c}(\bar{i})$.

Now, consider an edge e of $G(\mathbf{A})$ from $(x_i, \mathbf{u}(\bar{i}))$ to $(y_i, \mathbf{v}(\bar{i}))$ such that $y_i > x_i$. Then, we add $(m(x_i, \mathbf{v}(\bar{i})) - 1)\varphi(x_i)$ to the weight $w(e)$ for each such index i. We denote $\tilde{w}(e)$ for the modified weight.

We artificially define a new vertex t' and an edge f from $t = \Lambda(\mathbf{A})$ to t' with the weight $\tilde{w}(f) = \sum_{i=1}^{d}(m(t_i) - 1)\varphi(t_i)$, where t_i is the i-th entry of t (and hence the largest entry of the i-th row of $\mathbf{A}$), and $m(t_i)$ is the number of occurrence of t_i in the i-th row of $\mathbf{A}$.

We denote $\tilde{G}(\mathbf{A})$ for the obtained weighted directed acyclic graph.

Theorem 2 *The utility of the optimal solution for the utility maximization problem equals the weight of the maximum weight path from* s *to* t' *of* $\tilde{G}(\mathbf{A})$*.*

Proof: Consider the sequence $C : \mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_k$ of columns giving the optimal arrangement maximizing the utility. A column $\mathbf{c}_i$ is called *pause* if $\Lambda(C_{i-1}) = \Lambda(C_i)$, where C_i is the matrix consisting of columns $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_i$. We consider the corresponding path p on $\tilde{G}(\mathbf{A})$ to the column sequence C ignoring all pause columns. Suppose that the element x of the i-th row of $\mathbf{A}$ appears for the first time (in the i-th row of columns) in $\mathbf{c}_\ell$, and for the last time in $\mathbf{c}_r$ in the sequence C. Let $\mathbf{v} = \mathbf{c}_{r+1}$, and let $\mathcal{C}(x, \mathbf{v})$ be the set of all columns $\mathbf{c}$ such that $\mathbf{c}$ have x as the i-th entry and $\mathbf{c}(\bar{i}) \leq \mathbf{v}(\bar{i})$. Note that $|\mathcal{C}(x, \mathbf{v})| = m(x, \mathbf{v}(\bar{i}))$. Then, if there is any column in $\mathcal{C}(x, \mathbf{v})$ that is not used in the prefix of C up to $\mathbf{c}_r$, it should be moved just before $\mathbf{v}$, so that $\varphi(x)$ is added to the utility. Therefore, in the optimal sequence, $\varphi(x)$ is counted exactly $m(x, \mathbf{v}(\bar{i}))$ times. In the path p, $\varphi(x)$ is counted once when x first appears, and $m(x, \mathbf{v}(\bar{i})) - 1$ times when a larger elements than x first appears on the i-th row. Thus, the weight of the maximum weight path on $\tilde{G}(\mathbf{A})$ equals the maximum utility. □

Corollary 1 **MaxCA** *can be solved in* $O(n^{d+1})$ *time.*

For solving **MinCA** for a general $\mathbf{A}$ allowing multiplicities, we use the same graph $G(\mathbf{A})$ as the multiplicity-free case, and modify the weights as follows:

If e is an edge from $\mathbf{v} = (v_1, v_2, \ldots, v_d)$ to $\mathbf{v}' = \mathbf{v} \oplus \mathbf{c}$ corresponding to a column $\mathbf{c} = (c_1, c_2, \ldots, c_d)$, a index i is called *non-dominated* if $c_i \geq v_i$. Also, i is called *non-terminal* if c_i is not the maximum element in the i-th row of $\mathbf{A}$. The weight $w'(e)$ is defined by the sum of utilities $\varphi(c_i)$ for all non-dominated and non-terminal indices i. We have the following theorem.

Theorem 3 *The shortest path from* s *to* $t = \Lambda(\mathbf{A})$ *with respect to the weight function* w' *in* $G(\mathbf{A})$ *gives the solution of* **MinCA***, and it is computed in* $O(n^{d+1})$ *time.*

4 Binary cases and the hardness of MinCA and MaxCA

We show that both **MinCA** and **MaxCA** are NP-hard even if $\mathbf{A}$ is a binary matrix.

4.1 Binary MinCA and min-sum set covering

Let us consider a special case where the matrix $\mathbf{A}$ is binary, that is, each entry of $\mathbf{A}$ is either 0 or 1. The total ordering is given by $0 < 1$. We assume there is no all-zero nor all-one row. We define the utility function φ by $\varphi(0) = 1$ and $\varphi(1) = a$ for any fixed nonnegative number a. We denote **MinCA** and **MaxCA** for a binary $\mathbf{A}$ and the above φ by $\mathbf{BiMinCA}(\mathbf{A}, a)$ and $\mathbf{BiMaxCA}(\mathbf{A}, a)$, respectively. We show the min-sum set covering problem(**MSSC**) defined as follows can be formulated as $\mathbf{BiMinCA}(\mathbf{A}, a)$.

Consider a hypergraph $\mathcal{H} = (X, \mathcal{F})$, where $X = \{1, 2, \ldots, n\}$ is the vertex set and $\mathcal{F} = \{F_1, F_2, \ldots, F_d\}$ is a family of subsets (called hyperedges) of X satisfying that $\bigcup_{i=1}^{d} F_i = X$. Let $\sigma \in \mathcal{S}_n$ be a permutation, and we arrange the elements of X in a list $\sigma(1), \sigma(2), \ldots, \sigma(n)$. The index $\iota_\sigma(k)$ of a hyperedge F_k is the first j such that $\sigma(j) \in F_k$. **MSSC**is the problem to find the permutation σ minimizing $\sum_{1 \le k \le d} \iota_\sigma(k)$.

In other words, **MSSC**minimizes the average cover time of hyperedges, while the ordinary minimum set covering problem minimizes the cover time of the latest hyperedge.

The **MSSC**is NP-hard . Moreover, its special version named the min-sum vertex covering (**MSVC**) for which the hypergraph is a graph is known to be NP-hard. Feige *et al.*[7] showed that **MSSC**is NP-hard to approximate within $(4 - \epsilon)$ approximation ratio. On the other hand, a greedy algorithm that selects the vertex contained in the largest number of uncovered edges gives a 4-approximation solution.

Given an instance $\mathcal{H}$ of min-sum set covering, we consider its incidence matrix $\mathbf{A}^{\mathcal{H}}$ such that its (i, j) entry $\mathbf{A}^{\mathcal{H}}(i, j)$ is 1 if and only if $j \in F_i$. We consider $\mathbf{BiMinCA}(\mathbf{A}^{\mathcal{H}}, a)$. Given a permutation σ, $\sigma\mathbf{A}^{\mathcal{H}}(k, \iota_\sigma(k))$ is the first 1 entry in the k-th row, and hence the row starts with $\iota_\sigma(k) - 1$ zero entries. Hence, $\Phi(\sigma\mathbf{A}^{\mathcal{H}}(k)) = (\iota_\sigma(k) - 1)\varphi(0) + |F_j|\varphi(1) = (\iota_\sigma(k) - 1) + a|F_j|$, since the first $\iota_\sigma(k) - 1$ zero entries and all the 1 entries contributes to the Φ value.

We denote $N_1(\mathbf{B})$ for the number of 1 entries of a matrix $\mathbf{B}$. By definition, $N_1(\mathbf{A}^{\mathcal{H}}) = \sum_{1 \le i \le d} |F_k|$. Therefore, $\Phi(\sigma\mathbf{A}^{\mathcal{H}}) = \sum_{1 \le k \le d} \iota_\sigma(k) - d + aN_1(\mathbf{A}^{\mathcal{H}})$. Since the second and third terms are irrelevant to the choice of σ, the optimal solution minimizing $\Phi(\sigma\mathbf{A}^{\mathcal{H}})$ gives the optimal permutation for the min-sum covering problem.

Theorem 4 $\mathbf{BiMinCA}(\mathbf{A}, a)$ *is NP-hard. Moreover, it is NP-hard to obtain a* $(4 - \epsilon)$*-approximation solution of* $\mathbf{BiMinCA}(\mathbf{A}, a)$ *for any* $\epsilon > 0$ *if* $a \le \frac{d}{N_1(\mathbf{A})}$*. On the other hand, if* $a \ge \frac{d}{N_1(\mathbf{A})}$*, the greedy algorithm attains a* 4*-approximation solution. In particular,* $\mathbf{BiMinCA}(\mathbf{A}, 0)$ *is not* $(4-\epsilon)$*-approximable, while* $\mathbf{BiMinCA}(\mathbf{A}, 1)$ *is* 4*-approximable.*

4.2 Utility maximization on binary matrices

Let us consider $\mathbf{BiMaxCA}(\mathbf{A}, 0)$. As we have seen, $\Phi(\sigma\mathbf{A}) = -n + \sum_{1 \le k \le d} \iota_\sigma(k)$. Since n is independent of the choice of σ, the optimal solution of $\mathbf{BiMaxCA}(\mathbf{A}, 0)$ maximize $\sum_{1 \le k \le d} \iota_\sigma(k)$, which we name the maximum-sum set covering problem (**MaxSSC**). This is the maximization version of the min-sum set covering problem which consider maximizing the average cover time of hyperedges. Especially, if the hypergraph is a graph, we call it the maximum-sum vertex covering problem (**MaxSVC**).

Theorem 5 **MaxSVC** *is NP-hard. Accordingly, both* **MaxSSC** *and* $\mathbf{BiMaxCA}(\mathbf{A}, 0)$ *are NP-hard.*

Proof: We reduce **MSVC** for a graph G to **MaxSVC** for its complement $\bar{G}$. For the complete graph on the vertex set of G, the objective function (the utility value) does not depend on the choice of the permutation σ, and let Y be its value. Let $\mathbf{A}(G)$ and $\mathbf{A}(\bar{G})$ are incidence matrices for G and $\bar{G}$, then $\Phi(\sigma\mathbf{A}(G)) + \Phi(\sigma\mathbf{A}(\bar{G})) = Y$. Thus, the permutation σ maximizing $\Phi(\sigma\mathbf{A}(\bar{G}))$ minimizes $\Phi(\sigma\mathbf{A}(G))$. □

5 Maximizing sum of lengths of maximum subsequences

So far, we have considered the first-fit non-decreasing subsequence in each row of the matrix. However it is not always the maximum utility subsequence of the row. Given a sequence $\mathbf{a}$, its maximum utility subsequence can be computed by dynamic programming easily. For the special case, if $\varphi \equiv 1$, the

problem is called the *longest non-decreasing subsequence problem*, and it is solved by using patience sorting in $O(n \log n)$ time[3].

We consider the following problems:

MaxUNDS: Given a matrix $\mathbf{A}$ and a utility function φ, find a permutation σ such that the sum of utilities of the maximum utility non-decreasing subsequences of rows of $\sigma\mathbf{A}$ is maximized.

MaxLNDS: Given a matrix $\mathbf{A}$, find a permutation σ such that the sum of lengths of longest non-decreasing subsequences of rows of $\sigma\mathbf{A}$ is maximized.

Example 5 *Let* $\mathbf{A} = \begin{pmatrix} \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{4} & \mathbf{5} & \mathbf{6} \\ 6 & \mathbf{2} & \mathbf{3} & 5 & \mathbf{4} & 1 \end{pmatrix}$. *Then the red numbers form the longest non-decreasing subsequences of rows with the total length* 9*, and the identity permutation gives a solution of* **MaxLNDS**.

If we consider the utility function $\varphi(x) = x$*, then* $\sigma\mathbf{A} = \begin{pmatrix} \mathbf{2} & \mathbf{3} & \mathbf{4} & \mathbf{5} & \mathbf{6} & 1 \\ \mathbf{2} & \mathbf{3} & \mathbf{5} & 4 & 1 & \mathbf{6} \end{pmatrix}$ *has a total utility* 36*, and this* σ *is a solution of* **MaxUNDS**.

MaxUNDS for a binary matrix and $\varphi(0) = 1$ and $\varphi(1) > n$ becomes an equivalent problem to the max-sum set covering problem, and hence it is NP-hard for a general d. Unfortunately, we do not know whether it is polynomial-time soluble even if $d = 2$.

For **MaxLNDS**, We have the following theorem:

Theorem 6 *(1)* **MaxLNDS** *for a general matrix* $\mathbf{A}$ *can be solved in* $O(n \log n)$ *time for* $d = 2$*.*
(2) **MaxLNDS** *for a multiplicity-free matrix* $\mathbf{A}$ *can be solved in* $O(n^4)$ *time for* $d = 3$*.*

For $d = 2$, the solution is depressingly simple. We consider the lexicographic ordering of columns so that $(a, b)^\top > (c, d)^\top$ if either $a > c$ or $a = c$ and $b > d$. Then, we sort the columns non-decreasingly according to the ordering to obtain the optimal solution of **MaxLNDS**. It takes $O(n/logn)$ time.

Indeed, suppose P is a longest nondecreasing subsequence of the first column of an optimal solution, and $L(P)$ is its length. We select an optimal solution that also maximizes $L(P)$ among the set of all optimal solutions. We assume $L(P) < n$ and give a contradiction. We insert any column outside P so that we have a column arrangement with a nondecreasing subsequence Q of length $L(P) + 1$ in the first row. This insertion may decrease the length of the longest nondecreasing subsequence of the second row at most 1, and hence this is also an optimal solution. This contradicts to the assumption. Hence, we have the first statement. We omit the proof for the case $d = 3$ in this version.

5.1 Minimizing the monotonicity

We may consider the problem of minimizing the sum of lengths of the longest increasing sequences of rows in a matrix. However, a more natural problem is to find a column permutation to minimize the total lengths of the longest monotone subsequences in rows of $\mathbf{A}$, where a monotone subsequence is either increasing or decreasing subsequence. Here, we assume that the matrix $\mathbf{A}$ is multiplicity-free.

Let us give a formal description. Given a multiplicity free sequence $\mathbf{a} = (a_1, a_2, \ldots, a_n)$, let $I(\mathbf{a})$ (resp. $D(\mathbf{a})$) be the length of the maximum increasing (resp. decreasing) subsequence of $\mathbf{a}$. We define the *monotonicity* of $\mathbf{a}$ by $\mu(\mathbf{a}) = \max\{I(\mathbf{a}), D(\mathbf{a})\}$. For considering the monotonicity, we can assume without loss of generality that $\mathbf{a}$ is a permutation of $\{1, 2, \ldots, n\}$.

Given a multiplicity-free $d \times n$ matrix $\mathbf{A}$, its monotonicity is $\mu(\mathbf{A}) = \sum_{i=1}^{d} \mu(\mathbf{A}_i)$, where $\mathbf{A}_i$ is the i-th row vector of $\mathbf{A}$. The min-sum monotonicity problem is to find a permutation $\sigma \in \mathcal{S}_n$ of columns minimizing $\mu(\sigma(\mathbf{A}))$.

Example 6 *Let* $\mathbf{A} = \begin{pmatrix} \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{4} & \mathbf{5} & \mathbf{6} \\ \mathbf{6} & 2 & 3 & \mathbf{5} & \mathbf{4} & \mathbf{1} \end{pmatrix}$ *with* $\mu(\mathbf{A}) = 10$*.* $\sigma\mathbf{A} = \begin{pmatrix} 5 & \mathbf{6} & 3 & 1 & \mathbf{4} & \mathbf{2} \\ \mathbf{4} & 1 & \mathbf{3} & 6 & 5 & \mathbf{2} \end{pmatrix}$ *attains* $\mu(\sigma\mathbf{A}) = 6$*, and it is the minimum.*

One important fact is that $\mu(\mathbf{a}) \geq \lceil\sqrt{n}\rceil$ by Erdős-Szekeres theorem [5]. More precisely, there is a one-to-one correspondence called Robinson-Schensted-Knuth correspondence [10] between the set of permutations and the set of pairs of *Young tableaux* of the same shapes (called *Young diagram*) , and $\mu(\mathbf{a})$ is the maximum of the width and height of the corresponding Young diagram, which is at least $\lceil\sqrt{n}\rceil$. There is an efficient algorithm to find the permutation corresponding to a given pair of Young tableaux. Moreover, we can easily count the number of permutations such that $\mu(\mathbf{a}) = \lceil n\rceil$.

Therefore, $\mu(\sigma(\mathbf{A})) \geq d\lceil\sqrt{n}\rceil$. On the other hand, if we consider a random permutation σ, the standard probabilistic argument gives the following:

Theorem 7 *If σ is randomly chosen, the expected length of the longest increasing subsequence of a row of $\mathbf{A}$ is bounded by $(1+\epsilon)e\sqrt{n}$ for any positive constant ϵ if n is sufficiently large. Here, e is the base of the natural logarithm.*

Proof: Since σ is random, each row of $\mathbf{A}$ is a random permutation. Let us consider a subsequence of length $K \geq \lceil ce\sqrt{n}\rceil$ for $c > 1 + \epsilon/2$, then the probability that the sequence is an increasing sequence is $\frac{1}{K!}$. The number of length K subsequence in a row is ${}_nC_K$, and hence there are ${}_nC_K$ such subsequences in the row. Thus, the expected number Z of length K increasing subsequences in a row is $Z = \frac{{}_nC_K}{K!} = \frac{n!}{(n-K)!(K!)^2} \leq \frac{n^K}{(K!)^2}$. By using Stirling's formula, $K! \geq \sqrt{2\pi K}K^K e^{-K}$, and hence $Z \leq (2\pi K)^{-1}(\frac{ne^2}{K^2})^K \leq (2e\pi\sqrt{n})^{-1}c^{-2ce\sqrt{n}}$. This is smaller than $\frac{1}{\sqrt{n}}$ if $2e\pi < c^{2ce\sqrt{n}}$, which we can assume to be true if n is sufficiently large. Thus, with a probability $1 - \frac{1}{\sqrt{n}}$, there is no increasing subsequence longer than K in the row. We can similarly show that with a probablity $1\frac{1}{\sqrt{n}2^{2e\sqrt{n}}}$ there is no increasing subsequence longer than $2K$ in the row. Thus, the expected length of the longest increasing subsequence is bounded by $K + \frac{2K}{\sqrt{n}} + \frac{n}{\sqrt{n}2^{2e\sqrt{n}}}$. If n is sufficiently large, it is bounded by $(1+\epsilon)e\sqrt{n}$. □

Since $e = 2.718\ldots < 2.72$, we have the following:

Corollary 2 *The random sampling algorithm attains the approximation ratio* 2.72 *for the min-sum monotonicity problem if n is sufficiently large.*

6 Cumulative utility optimization

We would like to seek for possibility of designing an approximate algorithm for the column arrangement problem. However, it looks difficult to design a good approximation algorithm for **MaxCA**, and hence we consider another problem in which we consider a cumulative utility as the objective function.

In this subsection, we assume the utility function φ is monotone, that is, $\varphi(x) \geq \varphi(y)$ if $x > y$. We define the *cumulative utility* $\Psi(\mathbf{a}) = \sum_{i=1}^{n}\varphi(\mathrm{Max}(\mathbf{a}, \leq i))$ of a sequence $\mathbf{a}$. In other words, $\Psi(\mathbf{a})$ is the area (above the x-axis) below the non-decreasing function defined by $y = \max_{i\leq x}\varphi(a_i)$ in the range $1 \leq x \leq n+1$. We define $\Psi(\mathbf{A}) = \sum_{i=1}^{d}\Psi(\mathbf{A}(i))$, which we call the *cumulative utility* of the $d \times n$ matrix $\mathbf{A}$. We can observe that $\Psi(\sigma\mathbf{a})$ is the area below the non-decreasing step function (and above the horizon) defined by the greedy non-decreasing subsequence of $\sigma\mathbf{a}$, and it is minimized if $\sigma\mathbf{a}$ itself is a non-decreasing sequence. Thus, the minimization of $\Psi(\sigma\mathbf{a})$ is equivalent to the sorting, and we first consider the minimization problem that finds σ minimizing $\Psi(\sigma\mathbf{A})$ for a given matrix $\mathbf{A}$.

Example 7 *Consider the identity utility function $\varphi(x) = x$. Let* $\mathbf{A} = \begin{pmatrix} 1 & \mathbf{20} & 2 & 3 & 4 & 5 \\ \mathbf{20} & 1 & 2 & 3 & 5 & 4 \end{pmatrix}$. *The matrix* $\mathrm{Max}(\mathbf{A}_i, \leq j)$ $(i = 1, 2; 1 \leq j \leq 6)$ *shoowing the step functions is* $\begin{pmatrix} \mathbf{1} & \mathbf{20} & \mathbf{20} & \mathbf{20} & \mathbf{20} & \mathbf{20} \\ \mathbf{20} & \mathbf{20} & \mathbf{20} & \mathbf{20} & \mathbf{20} & \mathbf{20} \end{pmatrix}$, *and* $\Phi(\sigma\mathbf{A}) = 101 + 120 = 221$. *On the other hand, consider* $\sigma\mathbf{A} = \begin{pmatrix} \mathbf{2} & \mathbf{3} & \mathbf{4} & \mathbf{5} & 1 & \mathbf{20} \\ \mathbf{2} & \mathbf{3} & \mathbf{5} & 4 & \mathbf{20} & 1 \end{pmatrix}$. *The matrix showing the step functions is* $\begin{pmatrix} \mathbf{2} & \mathbf{3} & \mathbf{4} & \mathbf{5} & \mathbf{5} & \mathbf{20} \\ \mathbf{2} & \mathbf{3} & \mathbf{5} & \mathbf{5} & \mathbf{20} & \mathbf{20} \end{pmatrix}$, *and* $\Psi(\mathbf{A}) = 39 + 55 = 94$.

For a set S of column vector, consider the matrix $M(S)$ consisting of them, and consider its signature $\Lambda(M(S)) = \mathbf{v} = (v_1, v_2, \ldots, v_d)$. Let us define a function $f(S) = \sum_{i=1}^{d} \varphi(v_i)$. Then, we can observe that f is a monotone submodular function.

Now, we can observe that $\Psi(\sigma\mathbf{A}) = \sum_{i=1}^{n} f(S_i)$, where S_i is the set of the first i column vectors of $\sigma\mathbf{A}$, and its minimization is **MLOP** of the monotone submodular function f. Therefore, applying a result of Iwata *et al.* [9], we have the following.

Theorem 8 *There exists a polynomial time 2-approximation algorithm for finding the permutation minimizing the cumulative utility.*

For the maximization problem, consider the following algorithm.

Algorithm GREEDY: Repeat 1 and 2 until there is no remaining row;
1. Select and remove the column with the maximum utility entry;
2. Remove all rows whose maximum utility entries were in the removed column;

The permutation is given by the order of selection of columns. If we select less than n columns, we append remaining columns in an arbitrary fashion.

Theorem 9 *The approximation ratio of GREEDY for maximizing the cumulative utility is* $\frac{2n}{2n-d+1}$ *if* $n \geq d$ *and* $\frac{2d}{n+1}$ *if* $n < d$. *Here, the ratio is measured as* OPT/GRE, *where* OPT *is the optimal utility value and* GRE *is the utility value obtained by GREEDY.*

7 Concluding remarks

This paper is an initial work (as far as the authors know) on the column arrangement problem of a matrix considering non-decreasing subsequences in its rows. Several different models are considered and theoretical studies are given. By the nature of an initial study, several problems are left open. For example, we do not know whether **MaxUNDS**has a polynomial time algorithm even for $d = 2$.

References

[1] Y. Azar, L. Gamzu, X. Yin, Multiple Intents Re-ranking, *41st STOC*:669-678, 2009.

[2] E. Balas, The Prize Collecting Traveling Salesman Problem and Its Applications, Chapter 14 of *The Traveling Salesman Problem and Its Variations* (G. Gutin, A.P. Punnen, ed.), 663-695, Springer Verlag, 2007.

[3] S. Bespamyatnikh, M. Segal, Enumerating longest increasing subsequences and patience sorting, *Information Processing Letters* **76**: 7-11, 2000.

[4] T. Cormen, C. Leiserson, R. Rivest, C. Stein, *Introduction to Algorithms,* 3rd edition, MIT Press, 2009.

[5] P. Erdős, G. Szekeres, A combinatorial problem in geometry, *Compositio Mathematica,***2**: 463—470, 1935.

[6] S. Even, Y. Shiloah, NP-Completeness of Several Arrangement Problems, *Technical Report, Technion* 1975.

[7] U. Feige, L. Lovász, P. Tetali, Approximating Min Sum Set Cover, *Algorithmica,***40**: 219-234, 2004.

[8] M.R. Garey, D. S. Johnson, L. Stockmeyer, Some Simplified NP-Complete Graph Problems, *Theoretical Computer Science* **1**: 237-267, 1976.

[9] S. Iwata, P. Tetali, P. Tripathi, Approximating Minimum Linear Ordering Problems, *Proc. RANDOM 2012* :206-217, 2012.

[10] D. Knuth, *The Art of Computer Programming Vol.3, Sorting and Searching*, 2nd edition, Addion-Wesley, 1998.

[11] K. Mehlhorn, *Multi-dimensional Searching and Computational Geometry (Data Structures and Algorithms 3)*, ETACS Monographs on Theoretical Computer Science, Vol.3, Springer Verlag, 1984.

Matroid Intersection with Restricted Oracles

Kristóf Bérczi[1]

MTA-ELTE Matroid Optimization Res. Group
ELKH-ELTE Egerváry Research Group
Eötvös Loránd University
Budapest, Hungary
kristof.berczi@ttk.elte.hu

Tamás Király[1]

ELKH-ELTE Egerváry Research Group
Eötvös Loránd University
Budapest, Hungary
tamas.kiraly@ttk.elte.hu

Yutaro Yamaguchi[2]

Osaka University
Osaka, Japan
yutaro.yamaguchi@ist.osaka-u.ac.jp

Yu Yokoi[3]

National Institute of Informatics
Tokyo, Japan
yokoi@nii.ac.jp

Abstract: Matroid intersection is one of the most powerful frameworks of matroid theory that generalizes various problems in combinatorial optimization. Edmonds' fundamental theorem provides a min-max characterization for the unweighted setting, while Frank's weight-splitting theorem provides one for the weighted case. Several efficient algorithms were developed for these problems, all relying on the usage of one of the conventional oracles for both matroids; e.g., we can ask for the rank of a subset in each matroid, or whether a subset is independent or not in each matroid. In this study, we consider the tractability of the matroid intersection problem under restricted oracles answering only the sum/minimum/maximum of the ranks of a subset in two matroids or whether a subset is independent in both matroids or not.

Keywords: Matroid intersection, Tractability, Rank sum oracle, Minimum rank oracle, Maximum rank oracle, Common independence oracle

1 Introduction

A cornerstone of matroid theory is the efficient solvability of the matroid intersection problem introduced by Edmonds [5]. Efficient algorithms for weighted matroid intersection were developed subsequently by Edmonds [6], by Lawler [11, 12], and by Iri and Tomizawa [8]. The min-max duality theorem of Edmonds [5] for the unweighted matroid intersection problem was generalized by Frank [7] to the weighted case.

In order to design matroid algorithms and to analyze their complexity, it should be clarified how matroids are given. As the number of bases can be exponential in the size of the ground set, defining a matroid in an explicit form is inefficient. Rather than giving a matroid as an explicit input, it is usually assumed that one of the standard oracles is available, and the complexity of the algorithm is measured by the number of oracle calls and other elementary steps.

All previous studies on matroid intersection basically assume the availability of one of the standard oracles for both matroids; e.g., we can ask for the rank of a subset in each matroid, or whether a subset

[1]Research is supported by the Lendület Programme of the Hungarian Academy of Sciences – grant number LP2021-1/2021 and by the Hungarian National Research, Development and Innovation Office – NKFIH, grant numbers FK128673 and TKP2020-NKA-06.

[2]Research is supported by JSPS KAKENHI Grant Numbers 20K19743 and 20H00605, and by Overseas Research Program in Graduate School of Information Science and Technology, Osaka University.

[3]Research is supported by JST PRESTO Grant Number JPMJPR212B.

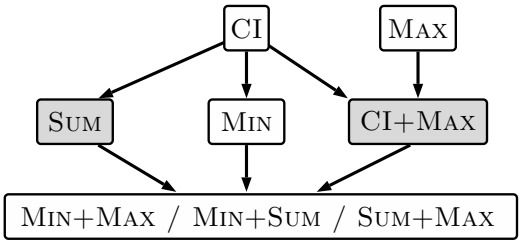


Figure 1: Hierarchy of oracles, where directed arcs representing polynomial reducibility, i.e., $A \to B$ means that A can be emulated by calling B polynomially many times. Grey boxes indicate oracles for which the tractability of weighted matroid intersection is settled.

is independent or not in each matroid. In this study, we ask if this assumption is really necessary for the tractability of matroid intersection.

One motivation comes from polymatroid matching, a framework introduced by Lawler [13] as a common generalization of matroid intersection and nonbipartite matching. In [15], Edmonds' theorem was deduced from polymatroid matching using a sophisticated argument. The main point is that when the matroid intersection problem is formulated as polymatroid matching, only the *rank sum function* of the two matroids is used rather than the two rank functions separately. Although the polymatroid matching problem cannot be solved in polynomial time in general [14, 9], the hardness was shown through special instances that seem to be far from matroid intersection. This suggests that matroid intersection might still be tractable when only the sum of the rank functions is available.

From the polyhedral viewpoint, the matroid intersection polytope (the convex hull of common independent sets) is determined by the *minimum rank function* [5]. It is known that the standard description of the polytope using the two rank functions separately enjoys nice properties such as total dual integrality and the tractability of the separation problem, but they may no longer be true if we describe it only by the minimum rank function. In an unpublished manuscript, Bárász [1] gave a polynomial-time algorithm for unweighted matroid intersection under the minimum rank oracle model, and it turns out challenging to extend the result to weighted matroid intersection.

2 Results

Our goal is to settle the tractability of the weighted matroid intersection problem under restricted oracles. In particular, we will focus on four different oracles: rank sum, minimum rank, maximum rank, and common independence oracles. The relation of the computational powers of combinations of these oracles is summarized as Figure 1.

The difficulty of giving an efficient algorithm is that the usual augmenting path approach cannot be applied directly, since the exchangeability graphs are not determined by restricted oracles. Still, using the rank sum oracle, we are able to give a strongly polynomial-time algorithm for the weighted matroid intersection problem by emulating the Bellman–Ford algorithm without explicitly knowing the underlying graph.

Theorem 1 *There exists a strongly polynomial-time algorithm for the weighted matroid intersection problem in the rank sum oracle model.*

The maximum rank oracle does not carry too much information on its own, as it cannot test the feasibility. However, when combined with the common independence oracle, they are strong enough to mimic our algorithm for the rank sum case.

Theorem 2 *There exists a strongly polynomial-time algorithm for the weighted matroid intersection problem when both the common independence and maximum rank oracles are available.*

For the common independence oracle, we have two tractable special cases as follows.

Theorem 3 *There exists a strongly polynomial-time algorithm for the unweighted matroid intersection problem in the common independence oracle model when one of the matroids is a partition matroid with all-one upper bound on the partition classes.*

Theorem 4 *There exists a strongly polynomial-time algorithm for the weighted matroid intersection problem in the common independence oracle model when one of the matroids is an elementary split matroid*[1]*.*

For the proofs, see [4].

3 Open problems

The following two big questions still remain, and are being tackled [2].

Question 5 *Is there a strongly polynomial-time algorithm for the weighted matroid intersection problem in the minimum rank model? Or can we show the hardness?*

Question 6 *Is there a strongly polynomial-time algorithm for the unweighted/weighted matroid intersection problem in the common independence oracle model? Or can we show the hardness?*

Acknowledgment

We are grateful to Yuni Iwamasa and Taihei Oki for initial discussions on the problem at HJ2019.

References

[1] M. Bárász. Matroid intersection for the min-rank oracle. Technical Report QP-2006-03, Egerváry Research Group, Budapest, 2006. `http://www.cs.elte.hu/egres/`.

[2] M. Bárász, K. Bérczi, T. Király, Y. Yamaguchi, and Y. Yokoi. Matroid intersection under minimum rank oracle. In preparation.

[3] K. Bérczi, T. Király, T. Schwarcz, Y. Yamaguchi, and Y. Yokoi. Hypergraph characterization of split matroids. *Journal of Combinatorial Theory, Series A*, 194:105697, 2023.

[4] K. Bérczi, T. Király, Y. Yamaguchi, and Y. Yokoi. Matroid intersection under restricted oracles. *SIAM Journal on Discrete Mathematics*, to appaer. (arXiv:2209.14516).

[5] J. Edmonds. Submodular functions, matroids, and certain polyhedra. In *Combinatorial Structures and Their Applications*, pages 69–87. Gorden and Breach, 1970. (Also in *Combinatorial Optimization — Eureka, You Shrink!*, pages 11–26, Springer, 2003.).

[6] J. Edmonds. Matroid intersection. *Annals of Discrete Mathematics*, 4:39–49, 1979.

[7] A. Frank. A weighted matroid intersection algorithm. *Journal of Algorithms*, 2(4):328–336, 1981.

[8] M. Iri and N. Tomizawa. An algorithm for finding an optimal "independent assignment". *Journal of the Operations Research Society of Japan*, 19(1):32–57, 1976.

[1] Recently, Joswig and Schröter [10] introduced the notion of split matroids, a class with distinguished structural properties that generalizes paving matroids. Bérczi, Király, Schwarcz, Yamaguchi and Yokoi [3] showed that every split matroid can be obtained as the direct sum of a so-called elementary split matroid and uniform matroids, and provided a hypergraph characterization of elementary split matroids.

[9] P. M. Jensen and B. Korte. Complexity of matroid property algorithms. *SIAM Journal on Computing*, 11(1):184–190, 1982.

[10] M. Joswig and B. Schröter. Matroids from hypersimplex splits. *Journal of Combinatorial Theory, Series A*, 151:254–284, 2017.

[11] E. L. Lawler. Optimal matroid intersections. In *Combinatorial Structures and Their Applications*, pages 233–234. Gorden and Breach, 1970.

[12] E. L. Lawler. Matroid intersection algorithms. *Mathematical Programming*, 9(1):31–56, 1975.

[13] E. L. Lawler. *Combinatorial Optimization: Networks and Matroids*. Holt, Rinehart and Winston, 1976.

[14] L. Lovász. The matroid matching problem. In *Algebraic Methods in Graph Theory II*, pages 495–517. North-Holland, 1981.

[15] L. Lovász and M. D. Plummer. *Matching Theory*. American Mathematical Society, 2009.

Orientation of convex sets

Péter Ágoston[1]

ELTE Eötvös Loránd University,
Budapest, Hungary
agostonp@cs.elte.hu

Gábor Damásdi[2]

ELTE Eötvös Loránd University,
Budapest, Hungary
damasdigabor@caesar.elte.hu

Balázs Keszegh[134]

Alfréd Rényi Institute of Mathematics and
ELTE Eötvös Loránd University,
Budapest, Hungary
keszegh@renyi.hu

Dömötör Pálvölgyi[13]

ELTE Eötvös Loránd University,
Budapest, Hungary
domotorp@gmail.com

Abstract: We introduce a novel definition of orientation on the triples of a family of pairwise intersecting planar convex sets and study its properties. In particular, we compare it to other systems of orientations on triples that satisfy a so-called interiority condition: $\circlearrowleft(ABD) = \circlearrowleft(BCD) = \circlearrowleft(CAD) = 1$ imply $\circlearrowleft(ABC) = 1$ for any A, B, C, D. We call such an orientation a P3O (partial 3-order), a natural generalization of a poset, that has several interesting special cases. For example, the order type of a planar point set (that can have collinear triples) is a P3O; we denote a P3O realizable by points as p-P3O.

If we do not allow $\circlearrowleft(ABC) = 0$, we obtain a T3O (total 3-order). Contrary to linear orders, a T3O can have a rich structure. A T3O realizable by points, a p-T3O, is the order type of a point set in general position. Despite these similarities to order types, P3O and T3O that can arise from the orientation of pairwise intersecting convex sets, denoted by C-P3O and C-T3O, turn out to be quite different from order types: there is no containment relation among the family of all C-P3O's and the family of all p-P3O's, or among the families of C-T3O's and p-T3O's.

A longer version of the paper can be found at arXiv:2206.01721.

Keywords: convex sets, order types, Helly-type theorems, set systems

1 Introduction

A family is *intersecting* if any two members of the family intersect, and it is *3-intersection-free* if no three members of the family have a common intersection. Such families of planar convex sets were studied by Jobson et al. [14] (see also Lehel and Tóth [16] and related recent results in extremal combinatorics

[1]This research has been implemented with the support provided by the Ministry of Innovation and Technology of Hungary from the National Research, Development and Innovation Fund, financed under the ELTE TKP 2021-NKTA-62 funding scheme.

[2]Supported by the ÚNKP-21-3 New National Excellence Program of the Ministry for Innovation and Technology from the source of the National Research, Development and Innovation fund.

[3]Supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences and by the ÚNKP-21-5 and ÚNKP-22-5 New National Excellence Program of the Ministry for Innovation and Technology from the source of the National Research, Development and Innovation Fund.

[4]Supported by the National Research, Development and Innovation Office – NKFIH under the grant K 132696 and FK 132060.

[17]). They showed that if three compact convex planar sets, A, B, C, form an intersecting and 3-intersection-free family, then $\mathbb{R}^2 \setminus (A \cup B \cup C)$ has exactly one bounded component, called the *hollow* of ABC, which we will denote by $\text{人}(ABC)$ (see Figure 1). They have also shown that the convex hull of this hollow is a triangle with sides a, b, c, such that (apart from its endpoints) side a is contained in $A \setminus (B \cup C)$, side b in $B \setminus (A \cup C)$, and side c in $C \setminus (A \cup B)$. We may refer to the vertices of this triangle as the *vertices of the hollow*, but note that since the hollow is open, its vertices are not a part of it, only of its closure.

From now on whenever we refer to a convex set, it is always assumed to be compact.

The following lemma is a straightforward consequence of Lemma 1 in [14].

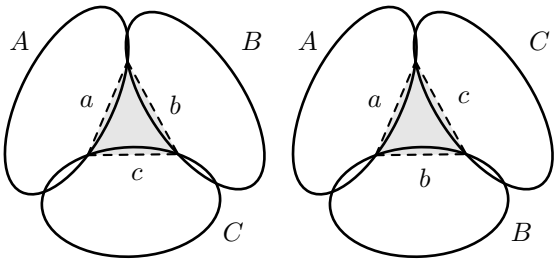


Figure 1: Three convex sets, A, B, C, with negative orientation on the left, and with positive orientation on the right, and their hollow, $\text{人}(ABC)$.

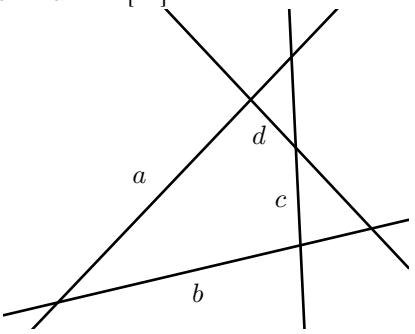


Figure 2: The following triples have positive orientation: *abc*, *abd*, *adc*, *bdc*.

Lemma 1 (Jobson-Kézdy-Lehel-Pervenecki-Tóth [14])**.** *Three pairwise intersecting compact convex sets, A, B, C, that do not have a common point, enclose a hollow $\text{人}(ABC)$, and the following four properties hold.*

(a) $\text{人}(ABC)$ is a simply connected region.

(b) The boundary of $\text{人}(ABC)$ has exactly one arc from each of the boundaries of A, B and C.

(c) The closure of the convex hull of $\text{人}(ABC)$ is a triangle with sides a, b, c such that (apart from its endpoints) side a is contained in $A \setminus (B \cup C)$, side b in $B \setminus (A \cup C)$, and side c in $C \setminus (A \cup B)$

(d) For any $x \in B \cap C$, $y \in A \cap C$ and $z \in A \cap B$ the orientation of the xyz triangle is the same and agrees with the orientation of abc.

Define the *orientation* of ABC, denoted $\circlearrowleft(ABC)$, as the orientation of the triangle with sides a, b, c: if the sides along the boundary of the triangle follow each other in a counterclockwise direction as abc, then we define the orientation of ABC as *positive*, otherwise as *negative*. We also define the orientation of three convex sets with a common intersection as *zero*. This way we can assign an orientation to any three members of an intersecting family of convex sets in the plane. To simplify notation, we assign a sign from $\{\pm 1, 0\}$ to each ordered triple according to their orientation and write $\circlearrowleft(ABC) = +1$, $\circlearrowleft(ABC) = -1$, $\circlearrowleft(ABC) = 0$, respectively, for positive, negative, zero orientations.[1] From the definitions, it follows that $\circlearrowleft(ABC) = \circlearrowleft(CAB) = \circlearrowleft(BCA) = -\circlearrowleft(ACB) = -\circlearrowleft(BAC) = -\circlearrowleft(CBA)$.

In general, we call $\{\pm 1, 0\}$ sign assignments to all triples of some base set satisfying the previous equalities *partial orientations*, and if the zero value is not allowed, *total orientations*. An orientation is called a *3-order* if it satisfies an extra property, called interiority condition, discussed in Section 2. Different 3-orders of interest are compared in Figure 6.

Remark 2. *Our definition only allows us to define an orientation for pairwise intersecting triples of convex sets. This is unlike the situation in the case of the (quite different) definition of orientation from*

[1] It might seem counterintuitive that the intersecting case is assigned 0 but this is the natural choice in some cases; see also [19, Section 4].

[4, 5, 6] by Bisztriczky and Fejes Tóth (later also investigated in [7, 8, 13, 18, 20, 21, 22]) which primarily focused on Erdős–Szekeres type theorems.[2] *In these papers the condition on the family of convex sets is that they are pairwise disjoint, or in later papers that they are non-crossing. Such a family is in convex position if no set is covered b the convex hull of the rest. In this case the orientation of ABC is determined by any points $a \in A, b \in B, c \in C$ chosen from the boundary of $conv(A \cup B \cup C)$. This definition appeared explicitly in [13] and is implicitly in earlier works—we will refer to it as the Bisztriczky–Fejes Tóth type orientation. Note that if A, B, C are in addition also intersecting but 3-intersection-free, then the Bisztriczky–Fejes Tóth type definition gives the same orientation as the one used in this paper. But such families can contain at most four connected sets, as K_5 is non-planar.*

If for some sets $A_1, \ldots, A_n$ we have $\circlearrowleft(A_i A_j A_k) = \delta$ for any $i < j < k$ for some $\delta \in \{\pm 1, 0\}$, then we will abbreviate this as $\circlearrowleft(A_1 \ldots A_n) = \delta$. With this notation, Helly's theorem says that if for some $n \geq 3$ planar convex sets $A_1, \ldots, A_n$ we have $\circlearrowleft(A_1 \ldots A_n) = 0$, then $\cap_{i=1}^n A_i \neq \emptyset$. We will also apply this shorthand notation for orientations of points, so for example $\circlearrowleft(abcd) = +1$ means that the points a, b, c, d are the vertices of a convex quadrangle, in this counterclockwise order. With this notation, Lemma 1(d) states $\circlearrowleft(xyz) = \circlearrowleft(ABC)$. Lemma 1(d) also implies the following.

Corollary 3. *If the convex sets A, B, C do not have a point in common, and the convex sets $A' \subset A, B' \subset B, C' \subset C$ are pairwise intersecting, then $\circlearrowleft(A'B'C') = \circlearrowleft(ABC)$.*

If a family of convex sets in the plane is intersecting and 3-intersection-free, we call it *holey*. For example, any collection of lines in general position is holey, and the orientation of any triple is determined by their slopes (see Figure 2). This orientation for lines is not to be confused with the much studied arrangement types of lines which were shown by Goodman and Pollack [10] to be the duals of order types of points, an order type meaning an orientation system belonging to a set of points in the plane, where all triples get the orientation based on whether they are in a counterclockwise or a clockwise order on their triangle. However, they also made the following simple observation about the orientations of triples of lines, which is relevant for us.

Claim 4 (Goodman-Pollack [11])**.** *If a holey family consists of lines $\ell_1, \ldots, \ell_n$, ordered according to their slopes in clockwise circular order, then the orientation of their triples is the same as the orientation of the triangles of n points $p_1, \ldots, p_n$ in convex position, ordered in counterclockwise order.*

Our main motivation to study holey families is that it can be the first step to improve our understanding of the intersection structure of planar convex sets, which can potentially lead to improved weak ε-nets [2] and (p, q)-theorems [3]. The question is, what abstract properties of the underlying geometric 3-hypergraphs are useful to derive interesting results.

In the following, we will call an orientation that can be derived from a holey family of planar convex sets a C-T3O (Convex 3-orders) and its superfamily where the sets are only required to be pairwise intersecting a C-P3O (Convex Partial 3-orders). Also, we call an order type belonging to points in the plane in general position a p-T3O or a *simple* order type, while an order type belonging to points in a plane in arbitrary position (allowing more than two points to be collinear) a p-P3O or a *partial* order type. The relationship between C-T3O's and C-P3O's is similar to the relationship between p-T3O's and p-P3O's. In Knuth [15] these are referred to as partial signings that can be completed to form order types.

The rest of this paper is organized as follows. In Section 2 we show that C-P3O's satisfy a natural interiority condition, and compare them with other well-studied orientations. In Section 3 we examine which p-T3O's (order types) are realizable as a C-T3O, and we find that up to five elements, the single condition that the configuration satisfies the interiority condition, is sufficient; this means that there exist C-T3O's that are not p-T3O's. We prove the strengthening that there is a p-T3O that is not a C-T3O in a companion paper [1], which primarily studies orientations of *good covers*, a generalization of the orientation studied here. Finally, in Section 4 we sketch some of our related results omitted from the paper and pose some open problems.

[2] For intersecting families, an Erdős–Szekeres type theorem with our definition of orientation follows directly from Ramsey's theorem.

2 Interiority

We say that a (partial) orientation satisfies the *interiority condition* if $\circlearrowleft(ABD) = \circlearrowleft(BCD) = \circlearrowleft(CAD) = 1$ imply $\circlearrowleft(ABC) = 1$ for any A, B, C, D. If $\circlearrowleft(ABD) = \circlearrowleft(BCD) = \circlearrowleft(CAD) = 1$ or $\circlearrowleft(ABD) = \circlearrowleft(BCD) = \circlearrowleft(CAD) = -1$ for some A, B, C, D, then we will write $D \in conv(ABC)$. (Note that the order of A, B, C is irrelevant in this notation.) This, however, can be quite misleading, as this notion of convexity does not have many natural properties, as we will see later.

Lemma 5 (Interiority Lemma)**.** *Any intersecting family of convex sets satisfies the interiority condition.*

Proof. Suppose A, B, C, O is an intersecting family of convex sets and $\circlearrowleft(ABO) = \circlearrowleft(BCO) = \circlearrowleft(CAO) = 1$. Then we need to show that $\circlearrowleft(ABC) = 1$.

For a contradiction, suppose first $\circlearrowleft(ABC) = 0$. Fix some $w \in A \cap B \cap C$, and take any $a \in A \cap O$, $b \in B \cap O$ and $c \in C \cap O$ and check the orientations of the triples of w, a, b, c using Lemma 1(d). We get $\circlearrowleft(abw) = \circlearrowleft(bcw) = \circlearrowleft(caw) = 1$. It follows that $w \in conv(a, b, c) \subset O$, contradicting that $\circlearrowleft(ABO) = 1$.

Now suppose $\circlearrowleft(ABC) = -1$. Take any $a \in A \cap O$, $b \in B \cap O$, $c \in C \cap O$, $z \in A \cap B$, $x \in B \cap C$ and $y \in A \cap C$. We can assume that these six points are in general position, otherwise we could slightly perturb them, along with the convex sets containing them, if necessary, without introducing a triple intersection. The conditions and Lemma 1(d) imply that $\circlearrowleft(abz) = \circlearrowleft(bcx) = \circlearrowleft(cay) = -1$ and $\circlearrowleft(xyz) = -1$. Also, as there is no triple intersection, we know that $x, y, z \notin conv(abc)$, $b, c, x \notin conv(ayz)$, $a, c, y \notin conv(bxz)$, $a, b, z \notin conv(cxy)$. We will deal with two cases, depending on the orientation of abc. The lines ab, bc, ca divide the plane into seven regions: a bounded triangle $conv(abc)$, three unbounded cones, which we denote by V_a, V_b, V_c, respectively, indexed by their apexes, and three unbounded regions sharing a side each with the triangle $conv(abc)$, which we denote by U_{ab}, U_{bc}, U_{ac}, respectively, indexed by the adjacent side of the triangle.

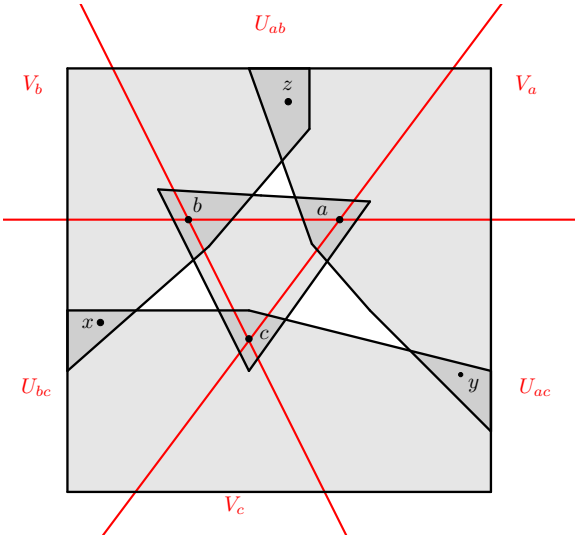


Figure 3: Case 1 of the proof of Lemma 5. Beware that in the figure $\circlearrowleft(xyz) = 1$ while in the proof $\circlearrowleft(xyz) = -1$ but we could find no better way to depict contradicting assumptions.

Case 1: $\circlearrowleft(abc) = 1$ (see Figure 3).
The orientation conditions and $x, y, z \notin conv(abc)$ imply that $x \in V_b \cup U_{bc} \cup V_c$, $y \in V_c \cup U_{ac} \cup V_a$, $z \in V_a \cup U_{ab} \cup V_b$.
Since $\circlearrowleft(xyz) = -1$, two of x, y, z must fall in the same cone V_i. Without loss of generality, assume that $x, y \in V_c$. As $c \notin conv(ayz)$, and a is to the right of the directed line yc, z must either lie to the right of line yc or to the left of the line ac. Since z lies to right of the line ab, if it lies to the left of ac then it is

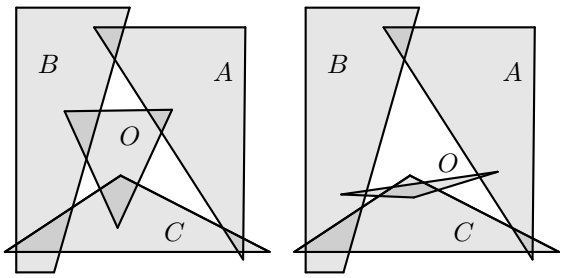


Figure 4: Regular and irregular containment $O \in conv(ABC)$.

in V_a. Hence z must lie to the right of yc. Similarly z must lie to the left of xc. But this implies $z \in U_{ab}$ and $\circlearrowleft(xyz) = 1$, a contradiction.

Case 2: $\circlearrowleft(abc) = -1$.

The orientation conditions and $x, y, z \notin conv(abc)$ imply that $x \in U_{ab} \cup V_a \cup U_{ac}$, $y \in U_{bc} \cup V_b \cup U_{ab}$, $z \in U_{ac} \cup V_c \cup U_{bc}$.

If any of x, y, z fall in a cone V_i, e.g., x falls in V_a then $y, z \in U_{a,c}$ and we can finish with a similar argument as in the previous case.

Otherwise, say that a has an *opposite* point, if $y \in U_{bc}$ or $z \in U_{bc}$ and, similarly, b has an opposite point, if $x \in U_{ac}$ or $y \in U_{ac}$ and c has an opposite point, if $x \in U_{ab}$ or $y \in U_{ab}$. If a does not have an opposite point, then $y \in U_{ab}$ and $z \in U_{ac}$, which implies that both b and c have an opposite point. Therefore, at least two of a, b, c have an opposite point, say, b and c. But then the segments connecting b and c to their opposite points intersect inside $conv(abc)$, which gives a triple intersection, contradicting our assumptions. □

Regular and irregular containment

In Figure 4 we can see two different ways $O \in conv(ABC)$ can happen. We will see that the one on the right complicates many scenarios, so we will often handle the two cases separately. We say that the containment $O \in conv(ABC)$ is *regular* if each of $O \cap \partial\text{人}(ABC) \cap A$, $O \cap \partial\text{人}(ABC) \cap B$ and $O \cap \partial\text{人}(ABC) \cap C$ is a connected set, and we say that the containment $O \in conv(ABC)$ is *irregular* if one of them has more than one connectivity component. If $O \in conv(ABC)$ is regular, then each of $O \cap \text{人}(ABC) \cap \partial A$, $O \cap \text{人}(ABC) \cap \partial B$ and $O \cap \text{人}(ABC) \cap \partial C$ is a connected curve.

But "doubly irregular containments" are impossible:

Claim 6. *For convex sets A, B, C and O, it is impossible that $O \in conv(ABC)$ and the containment is irregular with respect to both A and B.*

We omit the proof due to the space restrictions.

Relation to other notions of orientation

Knuth [15] studied orientations that satisfy the interiority condition under the name *interior triple system*, according to Knuth "for want of a better name." We want a better name, so we will refer to such an orientation as a T3O (total 3-order), while if zero-orientations are also allowed, then we call such an orientation a P3O (partial 3-order). We believe that these names are better as they reflect the similarity to posets, which would be called a P2O (partial 2-order) in our language. A poset can be considered a mapping from the ordered pairs of its base sets to $\{\pm 1, 0\}$ requiring antisymmetry and transitivity. Similarly, a P3O does the same for ordered triples, but in our case requiring the interiority condition.

Lemma 5 implies that the orientation of the triples of any holey family is a T3O. To the best of our knowledge, T3O's have not been studied anywhere except [15, Chapter 3], where the main result is that there are $2^{\Omega(n^3)}$ different T3O's over n elements.

If we add another property, called *transitivity* (the definition of which we omit here), then we get a much better studied notion, known as CC systems [15] (see also pseudoline arrangements [12, Chapter 5] and acyclic rank 3 oriented matroids [12, Chapter 6]). The transitivity property, however, is not satisfied by holey convex families. In fact, not even the following weaker condition, that we define below.

Knuth [15, Chapter 2, (2.4)] defines the *interior transitivity* condition as follows: If $D \in conv(ABC)$ and $E \in conv(ABD)$, then $E \in conv(ABC)$. The interior transitivity condition is satisfied by the earlier mentioned CC systems, but it is strictly weaker than them. Indeed, Knuth proved that the number of 3-orders on n sets is $2^{\Omega(n^2 \log n)}$, while the number of CC systems is $2^{\Theta(n^2)}$, and the number of CC systems that are representable by planar point sets, known as stretchable arrangements/order types, is $2^{\Theta(n \log n)}$. There are holey families that do not satisfy the interior transitivity condition, see Figure 5.

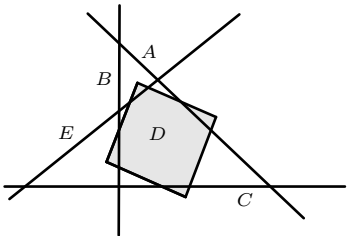


Figure 5: A family of convex sets not satisfying the interior transitivity: $D \in conv(ABC)$ and $E \in conv(ABD)$ but $E \notin conv(ABC)$.

However, the following weaker statement is true.

Claim 7. *If A, B, C, D and E are convex sets forming a holey family such that $D \in conv(ABC)$ and $E \in conv(ABD)$, then $D \cap E \subset \text{人}(ABC)$.*

For the proof we need the following simple observation which follows from checking how a regular or irregular containment can look like; see Figure 4.

Claim 8. *Suppose A, B, C and O are elements of a holey family.*
Then $O \in conv(ABC)$ if and only if $\text{人}(ABO)$, $\text{人}(BCO)$, $\text{人}(CAO) \subset \text{人}(ABC) \cup A \cup B \cup C$.

Proof of Claim 7. Since $D \cap E$ intersects $\partial\text{人}(ABD)$ which is contained in $\text{人}(ABC) \cup A \cup B \cup C$ by Claim 8, and $D \cap E$ cannot intersect $A \cup B \cup C$ as there are no triple intersections, we get that $D \cap E \subset \text{人}(ABC)$, as required. □

3 Small cases

Here we examine which T3O's on few elements are realizable with a holey family of convex sets, similarly as was done in [9] for allowable sequences and order types. In case of 4 elements, it follows from Lemma 5 that all system definitions coincide:

Claim 9. *On four elements, there are two* p-T3O*'s, two* T3O*'s and two* C-T3O*'s.*

In case of 5 elements, a p-T3O is determined by the size of the convex hull of the realizing point set, which gives three options, but by enumeration, there are six combinatorially different T3O's. We could realize all of them with convex sets (see Figures 7 and 8) which implies:

[3]For the Bisztriczky-Fejes Tóth type definition of order types of convex sets, any point order type is by definition realizable by convex sets, while in the other direction a configuration of convex sets whose order type is not realizable by points was given in [21] answering a question of Hubard and Montejano.

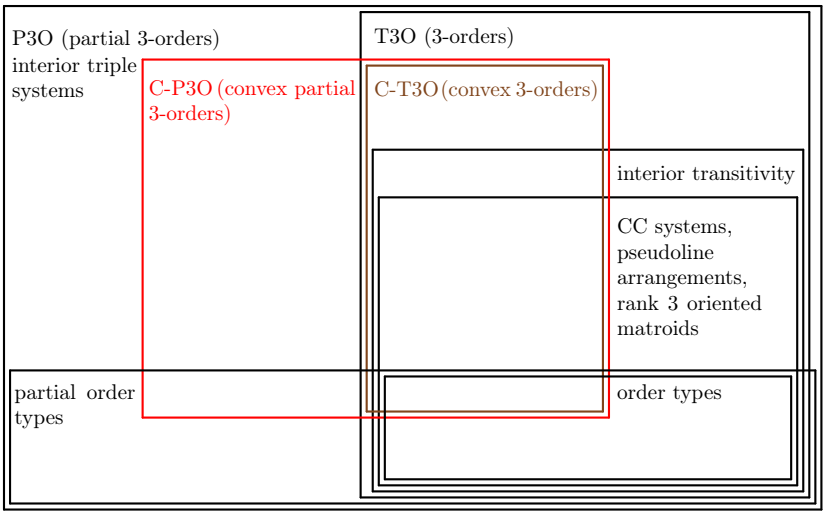


Figure 6: A diagram illustrating the relationship of some related notions. A P3O is any partial orientation of triples satisfying the interiority condition. A T3O is a P3O such that no triple is zero-oriented. A C-T3O (resp. C-P3O) is a T3O (resp. P3O) that is realizable with planar convex sets. Theorem **??** shows that a p-P3O might not be a C-P3O, while that a p-T3O might not be a C-T3O is proved in our companion paper [1], which also includes several further subclasses of P3O and T3O.[3]

Claim 10. *Any one of the six* T3O*'s on five elements is a* C-T3O*, i.e., it is representable by a holey family of convex sets.*

In case of 6 elements, it can be checked by enumeration that in total there are 253 combinatorially different T3O's on 6 elements, which is much more than 16, the number of p-T3O's realizable with 6 points.

We have managed to realize 14 out of the 16 p-T3O's as C-T3O's, while we conjecture that the other two cannot be realized. The list of these realizations can be found in Figure 9.

The 11th point set was more difficult to realize than the others, since (as we proved it, but the proof is omitted due to space restrictions) it does not have a realization where A, B and C contain all of D, E and F regularly.

4 Closing remarks

We had to omit some results from this version of the paper due to space restrictions.

We have shown a four-element holey family that is non-extendible, meaning that it is not part of any larger holey family (Figure 10). The result is rather simple, but it shows an important difference between point configurations and holey families, as a planar point set in general position always can be extended to another one.

We proved that the order type in Figure 11 is not a C-P3O.

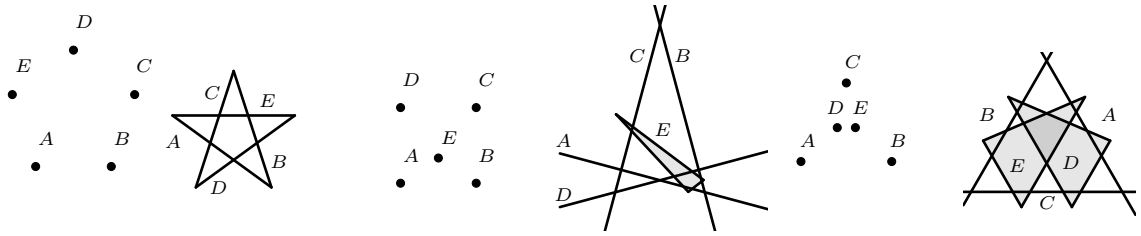


Figure 7: Three T3O's on five elements can be realized as a p-T3O and as a C-T3O.

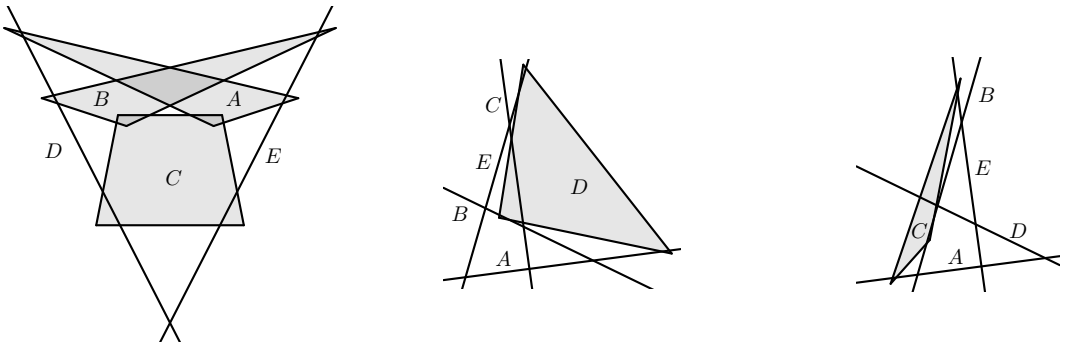


Figure 8: Three T3O's on five elements can be realized as a C-T3O but not as a p-T3O.

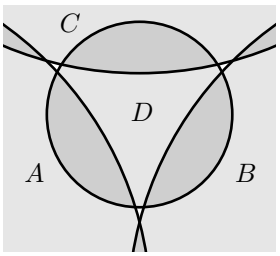


Figure 10: A holey family consisting of four disks that cannot be extended.

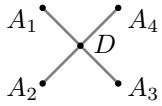


Figure 11: A p-P3O that is not a C-P3O.

We also examined a restricted case of the problem in which we take C-P3O's satisfying the (4,3)-property (all 4 sets contain an intersecting 3-tuple). We can prove that if we only use Lemma 5, two other observations about C-P3O's and the (4,3)-property, we still can find a hypergraph on n vertices, whose maximal clique size is $O(\sqrt{n})$. If the contrary would be true, it could help us creating new theorems about the intersections of convex sets.

In our next paper [1] (also submitted to this conference), we proved that not all p-T3O's are C-T3O's.

Our definition of orientation can be generalized to intersecting pseudo-disk arrangements and to $d+1$ convex sets in $\mathbb{R}^d$. We leave these for future research, like the following questions left open in this paper.

Problem 11. *Are all* C-P3O*'s and/or* C-T3O*'s extendable by adding one more element?*

Problem 12. *Are all 6-point order types* C-T3O*'s, or the two that we could not realize in Fig. 9 are not?*

Problem 13. *What further properties of* C-P3O*'s are needed to obtain efficient* (p, q) *theorems?*

Acknowledgments.
We would like to thank Nóra Frankl and Márton Naszódi for discussions during the project.

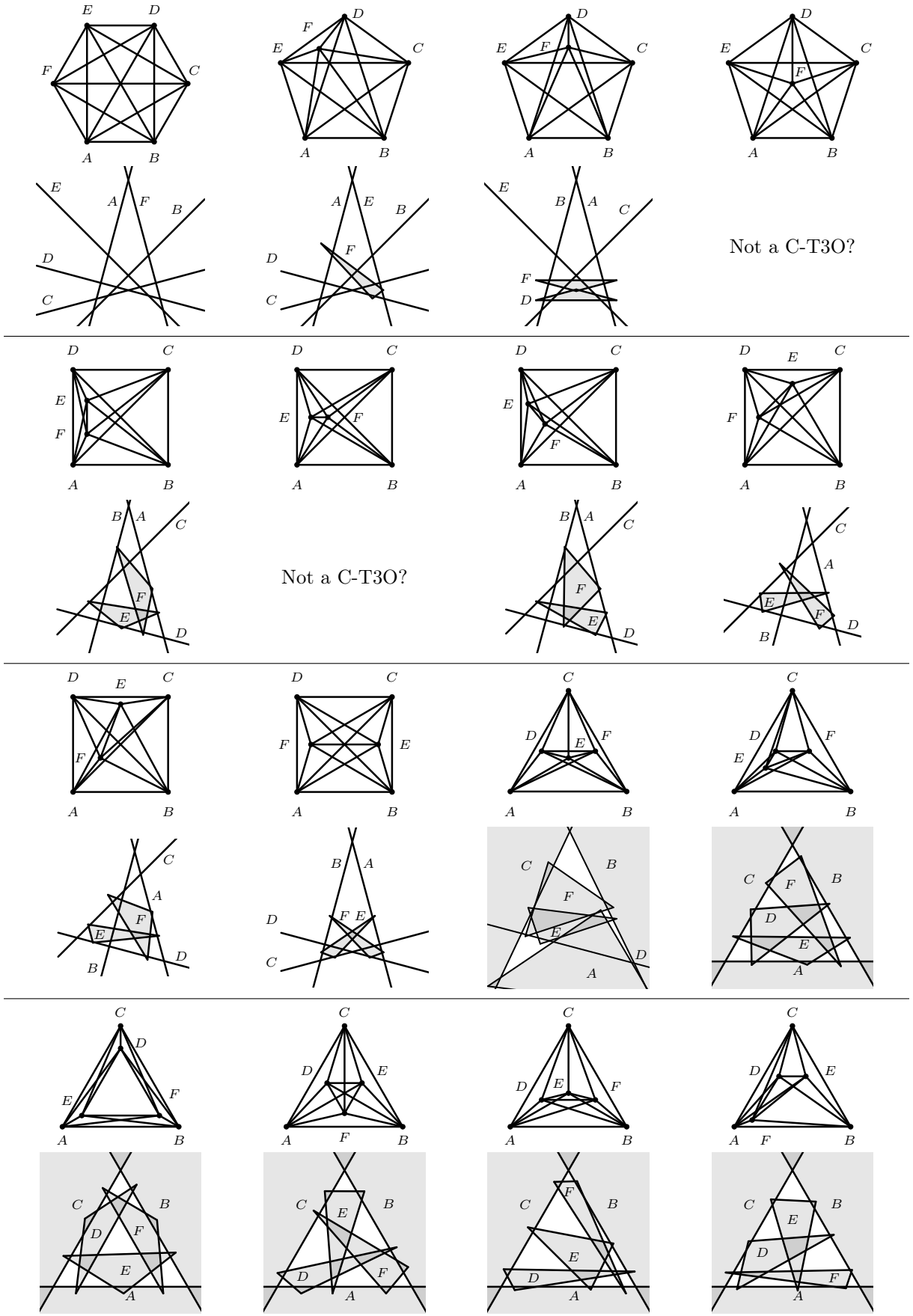


Figure 9: C-T3O's representing 6-point order types.

References

[1] P. Ágoston, G. Damásdi, B. Keszegh, and D. Pálvölgyi. Orientation of good covers. Preprint, `https://arxiv.org/abs/2206.01723`.

[2] N. Alon, I. Bárány, Z. Füredi, and D. J. Kleitman. Point selections and weak ε-nets for convex hulls. *Combinatorics, Probability and Computing*, 1(3):189–200, 1992.

[3] N. Alon and D. J. Kleitman. Piercing convex sets and the Hadwiger-Debrunner (p, q)-problem. *Advances in Mathematics*, 96(1):103–112, 1992.

[4] T. Bisztriczky and G. Fejes Tóth. A generalization of the Erdős-Szekeres convex n-gon theorem. *Journal für die Reine und Angewandte Mathematik*, 395:167–170, 1989.

[5] T. Bisztriczky and G. Fejes Tóth. Nine convex sets determine a pentagon with convex sets as vertices. *Geometriae Dedicata*, 31(1):89–104, 1989.

[6] T. Bisztriczky and G. Fejes Tóth. Convexly independent sets. *Combinatorica*, 10(2):195–202, 1990.

[7] M. G. Dobbins, A. Holmsen, and A. Hubard. The Erdős-Szekeres problem for non-crossing convex sets. *Mathematika*, 60(2):463–484, 2014.

[8] M. G. Dobbins, A. Holmsen, and A. Hubard. Regular systems of paths and families of convex sets in convex position. *Transactions of the American Mathematical Society*, 368(5):3271–3303, 2016.

[9] J. E. Goodman and R. Pollack. On the combinatorial classification of nondegenerate configurations in the plane. *Journal of Combinatorial Theory. Series A*, 29:220–235, 1980.

[10] J. E. Goodman and R. Pollack. A theorem of ordered duality. *Geometriae Dedicata*, 12:63–74, 1982.

[11] J. E. Goodman and R. Pollack. Semispaces of configurations, cell complexes of arrangements. *Journal of Combinatorial Theory. Series A*, 37:259–293, 1984.

[12] Handbook of Discrete and Computational Geometry (edited by J. E. Goodman, J. O'Rourke and Cs. D. Tóth), Third Edition, CRC Press LLC, Boca Raton, FL 2017.

[13] A. Hubard, L. Montejano, E. Mora, and A. Suk. Order types of convex bodies. *Order*, 28(1):121–130, 2011.

[14] A. S. Jobson, A. E. Kézdy, J. Lehel, T. J. Pervenecki, and G. Tóth. Petruska's question on planar convex sets. *Discrete Mathematics*, 343(9):1–13, 2020.

[15] D. E. Knuth. Axioms and hulls. Springer-Verlag, *Lecture Notes in Computer Science*, 1992.

[16] J. Lehel and G. Tóth. On the hollow enclosed by convex sets. *Geombinatorics*, 30(3):113–122, 2021.

[17] D. Nagy and B. Patkós. Triangles in intersecting families. Preprint, `https://arxiv.org/abs/2201.02452` 2022.

[18] J. Pach and G. Tóth. A generalisation of the Erdős-Szekeres theorem to disjoint convex sets. *Discrete & Computational Geometry*, 19(3):437–445, 1998.

[19] J. Pach and G. Tardos. Forbidden paths and cycles in ordered graphs and matrices. *Israel Journal of Mathematics*, 155:359–380, 2006.

[20] J. Pach and G. Tóth. Erdős-Szekeres-type theorems for segments and noncrossing convex sets. *Geometriae Dedicata*, 81(1-3):1–12, 2000.

[21] J. Pach and G. Tóth. Families of convex sets not representable by points. *Indian Statistical Institute Platinum Jubilee Commemorative Volume–Architecture and Algorithms*, 43–53, 2009.

[22] A. Suk. On order types of systems of segments in the plane. *Order*, 27(1):63–68, 2010.

An FPT algorithm for the envy-free ride allocation with respect to destination types

YUKI AMANO

Kyoto University, Japan
ukiamano@kurims.kyoto-u.ac.jp

AYUMI IGARASHI

University of Tokyo, Japan
igarashi@mist.i.u-tokyo.ac.jp

YASUSHI KAWASE

University of Tokyo, Japan
kawase@mist.i.u-tokyo.ac.jp

KAZUHISA MAKINO

Kyoto University, Japan
makino@kurims.kyoto-u.ac.jp

HIROTAKA ONO

Nagoya University, Japan
ono@nagoya-u.jp

Abstract: The model called *a fair ride allocation on a line* is proposed by Amano et al. as an extension of the airport problem to the so-called assignment setting, i.e., for multiple facilities and agents, each agent chooses a facility to use and shares the cost with the other agents. Such a situation can often be seen in sharing economy, such as sharing fees for office desks among workers, and taxi fare among customers of possibly different destinations on a line. In this study, we consider envy-freeness as the criteria of fairness and design an FPT algorithm for the envy-free ride allocation with respect to destination types in the model.

Keywords: Envy-freeness, Airport games, Cooperative games, Shapley value

1 Introduction

Imagine a group of university students, each of whom would like to take a taxi to her/his destination. For example, Alice wants to go directly back home, while Bob prefers to go downtown to meet with friends. Each student may ride a taxi alone or share a ride and split into multiple groups to benefit from sharing the cost. It is then natural to ask two problems: how to form coalitions and fairly divide the fee.

In order to handle such a situation, the model called *a fair ride allocation on a line* is proposed by Amano et al. [2] as an extension of the airport problem. The *airport problem* is a classical fair division problem introduced by Littlechild and Owen [15] in which it is decided how to distribute the cost of a facility among agents when the greatest demand of the agents determines the cost. The model of Amano et al. [2] is an extension of the airport problem to the so-called assignment setting, i.e., for multiple facilities and agents, each agent chooses a facility to use and shares the cost with the other agents. Here, the facilities are taxis, the agents' demands are their destinations, and the costs are the fare of taxis. In the model, there are n agents A and k taxis, and each agent rides a taxi at the same initial location of the point 0. Each agent $a \in A$ has destination $x_a \in \mathbb{R}_{>0}$, which is called the *destination type* of agent a. Each taxi i has a quota q_i representing its capacity. We consider allocating all agents to taxis subject to the quota constraints of all taxis. The agents assigned to each taxi i, denoted by $T_i \subseteq A$, is charged a cost according to the furthest destination $\max_{a \in T_i} x_a$, and each agent pays one's share of the cost. The Shapley value is used as the payment rule, where the rule's definition can be found in Section 2. Note that the model with one taxi without quota constraint is identical to the airport problem. Since the model is a natural generalization of

the airport problem, it can be applied in various situations such as sharing office rooms, traveling along a highway, and boat traveling on a river; see details in Thomson [23].

In this paper, we consider envy-freeness [12], which requires that no agent prefers to replace another agent, as the fairness of allocation. We show that an envy-free allocation for the model can be computed in FPT time with respect to the number of destination types. More precisely, when the number p of destination types is small, we enumerate all possible 'shapes' of envy-free allocations by utilizing the monotonicity and the split property, and then compute an envy-free feasible allocation in $O(p^p n^4)$ time by exploring semi-lattice structure of size vectors consistent with a given shape. Note that the result has already been mentioned by Amano et al. [2]. In this paper, we present an algorithm for finding such an allocation and give a rough proof of the validity of the algorithm. Efficient algorithms are known for finding a feasible envy-free allocation when the number of taxis or the capacity of each taxi is small Amano et al. [2], where they are based on three structural properties of envy-free allocations: *monotonicity*, *split property*, and *locality*. On the negative side, they showed that it is NP-hard to decide if there exists an allocation under two relaxed envy-free concepts [2]. The first one relaxes the envy-free requirement by imposing the necessary conditions in Split Lemma (split conditions). It is NP-complete to decide whether there exists a feasible allocation that satisfies the split conditions. The second one generalizes the notion of envy-freeness by looking into envies among particular ordered pairs. For a subset $S \subseteq A^2$ of the agents' ordered pairs, an allocation is called *envy-free in* S if agent a never envies agent b for any $(a, b) \in S$. Given a subset $S \subseteq A^2$, it is also NP-complete to decide whether there exists a feasible allocation that is envy-free in S.

Related work

The problem of fairly dividing the cost among multiple agents has been long studied in the context of cooperative games with transferable utilities; we refer the reader to the book of Chalkiadakis et al. [9] for an overview. Following the seminal work of Shapley [22], several researchers have investigated the axiomatic property of the Shapley value as well as its applications to real-life problems. Littlechild and Owen [15] analyzed the property of the Shapley value when the cost of each subset of agents is given by the maximum cost associated with the agents in that subset. Chun and Park [10] analyzed the extension of the airport problem under capacity constraints wherein the number of facilities is unlimited and the capacity of each facility is the same. They showed that a capacity-adjusted version of the sequential equal contributions rule coincides with the Shapley value of a cooperative game whose characteristic value is defined as the minimum cost of serving all the members of a coalition. Note that the payment rule in Chun and Park [10] is different from our setting: they consider a global payment rule that divides the total cost among all agents, while in our setting, the cost of each facility is divided locally among the agents assigned to the same facility. CHUN et al. [11] further studied the strategic process in which agents divide the cost of the resource, showing that the division by the Shapley value is indeed a unique subgame perfect Nash equilibrium under a natural three-stage protocol.

Our work is similar in spirit to the complexity study of congestion games [16, 19]. In fact, without capacity constraints, it is not difficult to see that the fair ride-sharing problem can be formulated as a congestion game. The fairness notions, including envy-freeness in particular, have been well-explored in the fair division literature. Although much of the focus is on resource allocation among individuals, several recent papers study the fair division problem among groups [14, 21]. Our work is different from theirs in that agents' utilities depend not only on allocated resources, but also on the group structure.

In the context of hedonic coalition formation games, e.g., Aziz and Savani [4], Barrot and Yokoo [6], Bodlaender et al. [7], Bogomolnaia and Jackson [8], there exists a rich body of literature studying fairness and stability. In hedonic games, agents have preferences over coalitions to which they belong, and the goal is to find a partition of agents into disjoint coalitions. While the standard model of hedonic games is too general to accommodate positive results (see Peters and Elkind [18]), much of the literature considers subclasses of hedonic games where desirable outcomes can be achieved. For example, Barrot and Yokoo [6] studied the compatibility between fairness and stability requirements, showing that top responsive games always admit an envy-free, individually stable, and Pareto optimal partition.

Finally, our work is related to the growing literature on the ride-sharing problem [1, 3, 5, 11, 13, 17, 20, 24]. Santi et al. [20] empirically showed a large portion of taxi trips in New York City could be shared while keeping prolonged passenger travel time low. Motivated by an application to the ride-sharing platform, Ashlagi et al. [3]

considered the problem of matching passengers for sharing rides in an online fashion. However, they did not study the fairness perspective of the resulting matching.

2 Model

The model called *a fair ride allocation on a line* is proposed by Amano et al. [2] as an extension of the airport problem to the so-called assignment setting. For a positive integer $s \in \mathbb{Z}_{>0}$, we write $[s] = \{1, 2, \ldots, s\}$. For a set T and an element a, we may write $T + a = T \cup \{a\}$ and $T - a = T \setminus \{a\}$. In our setting, there are a finite set of *agents*, denoted by $A = [n]$, and a finite set of k *taxis*. The nonempty subsets of agents are referred to as *coalitions*. Each agent $a \in A$ is endowed with a destination $x_a \in \mathbb{R}_{>0}$, which is called the *destination type* (or shortly *type*) of agent a. We assume that the agents ride a taxi at the same initial location of the point 0, and they are sorted in nondecreasing order of their destinations, i.e., $x_1 \leq x_2 \leq \cdots \leq x_n$. Each taxi $i \in [k]$ has a quota q_i representing its capacity, where $q_1 \geq q_2 \geq \cdots \geq q_k$ (> 0) is assumed. An *allocation* $\mathcal{T} = (T_1, \ldots, T_\ell)$ is an ordered partition of A, and is called *feasible* if $\ell \leq k$ and $|T_i| \leq q_i$ for all $i \in [\ell]$. Given a monotone nondecreasing function $f\colon \mathbb{R}_{>0} \to \mathbb{R}_{>0}$, the *cost* charged to agents in T_i is the value of f in the furthest destination $\max_{a \in T_i} f(x_a)$ if $|T_i| \leq q_i$, and ∞ otherwise. The cost has to be divided among the agents in T_i. Without loss of generality, we assume that the cost charged to T_i is simply the distance of the furthest destination if $|T_i| \leq q_i$, i.e., f is the identity function. In other words, we may regard that x_a is the cost itself instead of the distance.

We consider a scenario where agents divide the cost using the well-known *Shapley value* [22], which, in our setting, coincides with the sequential contributions rule as in the airport problem [15]. Formally, for each subset T of agents and $s \in \mathbb{R}_{>0}$, we denote by $n_T(s)$ the number of agents a in T whose destinations x_a is at least s, i.e., $n_T(s) \coloneqq \big|\{a \in T \mid x_a \geq s\}\big|$. For each coalition $T \subseteq A$ and positive real $x \in \mathbb{R}_{>0}$, we define

$$\varphi(T, x) = \int_0^x \frac{\mathrm{d}r}{n_T(r)},$$

where we define $\varphi(T, x) = \infty$ if $n_T(x) = 0$. Equivalently, for a subset T of s agents whose destinations are given by $x_{i_1} \leq x_{i_2} \leq \ldots \leq x_{i_s}$, the value $\varphi(T, x_{i_j})$ is given by $x_{i_1}/s + (x_{i_2} - x_{i_1})/(s-1) + \ldots + (x_{i_j} - x_{i_{j-1}})/(s-j+1)$. See Example 1 for an illustration of the sequential contributions rule.

Throughout, we use a succinct notation to specify examples. An instance will be denoted as a single arrow where the black circles on each arrow will denote the set of agents who drop off at the same destination. An allocation $\mathcal{T}$ is written as a set of arrows where the arrows correspond to coalitions $T \in \mathcal{T}$, and the black circles on the arrow T denote the set of destinations of the agents in T.

Example 1 *Consider a taxi that forms a coalition T in Fig. 1, i.e., agents a, b, c, and d take one taxi together from a starting point to points 12, 24, 36, and 40 on a line, respectively. The total cost is 40, which corresponds to the drop-off point of d. According to the payment rule, agents a, b, c, and d pay 3, 7, 13, and 17, respectively. In fact, from the starting point to the drop-off point of a, To see this, observe that all the agents are in the taxi from the starting point to the drop-off point of a, so they equally divide the cost of 12, which means that a should pay $\varphi(T, x_a) = 12/4 = 3$. Between the dropping points of a and b, three agents are in the taxi, so they equally divide the cost of $24 - 12 = 12$, resulting in the cost of 4 for each of the three agents. Thus, $\varphi(T, x_b) = 12/4 + 12/3 = 3 + 4 = 7$. By repeating similar arguments, we have $\varphi(T, x_c) = 7 + (36 - 24)/2 = 13$, and $\varphi(T, x_d) = 13 + (40 - 36) = 17$.*

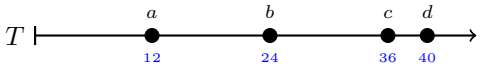


Figure 1: The coalition in Example 1

For an allocation $\mathcal{T}$ and a coalition $T_i \in \mathcal{T}$, the *cost* of agent $a \in T_i$ is defined as $\Phi_{\mathcal{T}}(a) \coloneqq \varphi_i(T_i, x_a)$ where

$$\varphi_i(T_i, x) = \begin{cases} \varphi(T_i, x) & \text{if } |T_i| \leq q_i, \\ \infty & \text{if } |T_i| > q_i. \end{cases}$$

It is not difficult to verify that the sum of the payments in T_i is equal to the cost of taxi i. Namely, if $|T_i| \leq q_i$, we have $\sum_{b \in T_i} \varphi_i(T_i, x_b) = \max_{a \in T_i} x_a$. On the other hand, if $|T_i| > q_i$, all agents in T_i pay ∞ whose sum is equal to ∞ (i.e., the cost of taxi i). It is already shown that the payment rule for each taxi coincides with the Shapley value [2].

3 Envy-free allocations

In this section, we consider envy-free feasible allocations for a fair ride allocation on a line. Note that there is a simple example with no envy-free feasible allocation as following Example 2. We thus study the problem of deciding the existence of an envy-free feasible allocation and finding one if it exists. We show that the problem is FPT with respect to the number of destination types.[1] This restriction is relevant to consider a setting where the number of destinations is small; for instance, a workshop organizer may offer a few excursion opportunities to the participants of the workshop.

We first give the formal definition of envy-free allocation and then describe three basic properties of envy-free allocations that will play important roles in designing of the algorithm in this paper.

Envy-freeness requires that no agent prefers another agent. Formally, for an allocation $\mathcal{T}$, agent $a \in T_i$ *envies* $b \in T_j$ if a can be made better off by replacing herself by b, i.e., $i \neq j$ and $\varphi_j(T_j - b + a, x_a) < \varphi_i(T_i, x_a)$. A feasible allocation $\mathcal{T}$ is *envy-free (EF)* if no agent envies another agent. Without capacity constraints, i.e., $q_1 \geq n$, envy-freeness can be trivially achieved by allocating all agents to a single coalition T_1. Also, when the number of taxis is at least the number of agents, i.e., $k \geq n$, an allocation that partitions the agents into the singletons is envy-free.

Example 2 *Consider an instance where $n = 4$, $k = 2$, $q_1 = q_2 = 2$, $x_1 = 2$, and $x_2 = x_3 = x_4 = 4$. We show that no feasible allocation is envy-free. To see this, let $\mathcal{T} = (T_1, T_2)$ be a feasible allocation. By feasibility, the capacity of each taxi must be full, i.e., $|T_1| = |T_2| = 2$. Suppose without loss of generality that $T_1 = \{1, 2\}$ and $T_2 = \{3, 4\}$ in Fig. 2. Then agent 2 envies the agents of the same type. Indeed, she needs to pay the cost of 3 at the current coalition while she would only pay 2 if she were replaced by 3 (or 4). Hence this instance has no envy-free feasible allocation.*

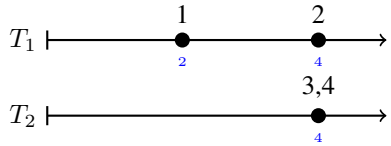


Figure 2: An instance with no envy-free feasible allocation

The first property is *monotonicity* of the size of coalitions in terms of the first drop-off point, which is formalized as follows.

Lemma 3 (Monotonicity lemma [2]) *For an envy-free feasible allocation $\mathcal{T}$ and non-empty coalitions $T, T' \in \mathcal{T}$, we have the following implications:*

$$\min_{a \in T} x_a < \min_{a' \in T'} x_{a'} \quad \textit{implies} \quad |T| \geq |T'|, \tag{1}$$

$$\min_{a \in T} x_a = \min_{a' \in T'} x_{a'} \quad \textit{implies} \quad |T| = |T'|. \tag{2}$$

[1] A problem is said to be *fixed parameter tractable* (FPT) with respect to a parameter p if each instance I of this problem can be solved in time $f(p) \cdot \mathrm{poly}(|I|)$.

We next describe the *split* property of envy-free feasible allocations. For a coalition T and a real s, we use notations $T_{<s}$, $T_{=s}$, and $T_{>s}$ to denote the set of agents with type smaller than s, equal to s, and larger than s, respectively. We say that *agents of type x are split in an allocation $\mathcal{T}$* if $\mathcal{T}$ contains two distinct T and T' with $T_{=x}, T'_{=x} \neq \emptyset$. The next lemma states that the agents of type x can be split in an envy-free feasible allocation only if they are the first passengers to drop off in their coalitions, and such coalitions are of the same size; further, if two taxis have an equal number of agents of split type, then no other agent rides these taxis.

An implication of the lemma is critical: we do not have to consider how to split the agents of non-first drop-off points to realize envy-free feasible allocations.

Lemma 4 (Split lemma [2]) *If agents of type x are split in an envy-free feasible allocation $\mathcal{T}$, i.e., $T_{=x}, T'_{=x} \neq \emptyset$ for some distinct $T, T' \in \mathcal{T}$, then we have the following three statements:*

(i) *The agents of type x are the first passengers to drop off in both T and T', i.e., $T_{<x} = T'_{<x} = \emptyset$,*

(ii) *Both T and T' are of the same size, i.e., $|T| = |T'|$, and*

(iii) *If $|T_{=x}| = |T'_{=x}|$, then $T = T_{=x}$ and $T' = T'_{=x}$.*

The last property of envy-free allocations is *locality*, i.e., every agent a is allocated to a taxi T with minimum cost $\varphi(T, x_a)$.

Lemma 5 (Locality lemma [2]) *For any envy-free allocation $\mathcal{T}$, coalition $T \in \mathcal{T}$, and agent $a \in T$, we have*

$$\varphi(T, x_a) \leq \varphi(T', x_a)$$

for all $T' \in \mathcal{T}$. Furthermore, the strict inequality holds if x_a is larger than the first drop-off point $\min_{a' \in T'} x_{a'}$ of T'.

4 FPT algorithm

In this section, we show that an envy-free feasible allocation can be computed in FPT time with respect to the number of destination types.

Recall that due to the split property, once we know the first drop-off points of each coalition, no agent of the other types will be split in an envy-free allocation. Thus, we can represent the 'shapes' of an envy-free allocation by a directed graph G where the first drop-off points can be considered as roots, followed by the agents of the other types. The main idea of our algorithm is to (1) enumerate all such G and (2) decide whether there is a size vector λ of each coalition that results in an envy-free outcome that is consistent with G. Although a naive approach to enumerate all possible size vectors gives rise to an $O(p^p n^p)$ algorithm where p is the number of destination types, we show that a more sophisticated approach results in an $O(p^p n^4)$ algorithm by utilizing structural properties of G and λ. In particular, we show that G and λ define a unique envy-free allocation (up to isomorphism), G is a star-forest, and λ forms semi-lattice.

Now, let $V = \{x_a \mid a \in A\}$ be the set of destination types, and let $p = |V|$. For an allocation $\mathcal{T} = (T_1, \ldots, T_k)$, we define its *allocation (di)graph* $G^{\mathcal{T}} = (V, E)$ by

$$E = \bigcup_{T \in \mathcal{T}} \left\{ (y, z) \in V^2 \;\middle|\; \begin{array}{l} y, z \in \{x_a \mid a \in T\}, y < z, \\ \nexists a \in T : y < x_a < z \end{array} \right\}.$$

Namely, the allocation graph $G^{\mathcal{T}}$ contains a directed edge (y, z) if and only if an agent of type y drops off just after an agent of type z in some coalition $T \in \mathcal{T}$. By definition, $G^{\mathcal{T}}$ is acyclic because every edge is oriented from a smaller type to a larger type, i.e., $(y, z) \in E$ implies $y < z$. We assume that all graphs discussed in this section satisfy the condition.

A graph is called a *star-tree* if it is a rooted (out-)tree such that all vertices except the root have out-degree at most 1, and a *star-forest* if each connected component is a star-tree. Then (i) in Split lemma implies that $G^{\mathcal{T}}$ is a star-forest. See the allocation graph for an envy-free feasible allocation is depicted in Fig. 3.

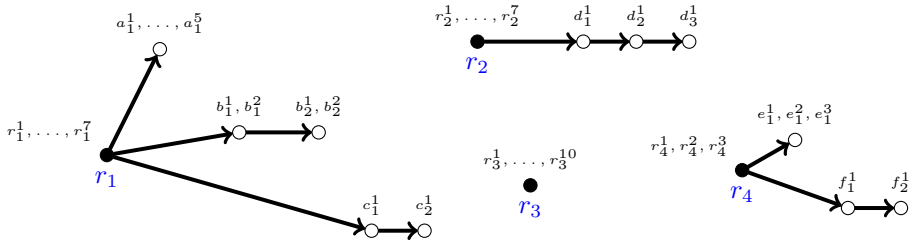


Figure 3: An example of the allocation graph for an envy-free feasible allocation $\mathcal{T} = (T_1, T_2, \ldots, T_9)$ where $T_1 = \{r_1^1, a_1^1, a_1^2, a_1^3, a_1^4, a_1^5\}$, $T_2 = \{r_1^2, r_1^3, b_1^1, b_1^2, b_2^1, b_2^2\}$, $T_3 = \{r_1^4, r_1^5, r_1^6, r_1^7, c_1^1, c_2^1\}$, $T_4 = \{r_2^1, r_2^2, d_1^1, d_2^1, d_3^1\}$, $T_5 = \{r_2^3, r_2^4, r_2^5, r_2^6, r_2^7\}$, $T_6 = \{r_3^1, r_3^2, r_3^3, r_3^4, r_3^5\}$, $T_7 = \{r_3^6, r_3^7, r_3^8, r_3^9, r_3^{10}\}$, $T_8 = \{r_4^1, e_1^1, e_1^2, e_1^3\}$, $T_9 = \{r_4^2, r_4^3, f_1^1, f_2^1\}$. There are seven agents of type r_1 ($r_1^1, \ldots, r_1^7$), seven agents of type r_2 ($r_2^1, \ldots, r_2^7$), ten agents of type r_3 ($r_3^1, \ldots, r_3^{10}$), and three agents of type r_4 (r_4^1, r_4^2, r_4^3).

Now, we explore the relationship between $\mathcal{T}$ and $G^{\mathcal{T}}$, implied by the Split lemma. Formally, let $\mathcal{C} = \{C_1, \ldots, C_t\}$ be the family of the vertex sets of connected components in $G^{\mathcal{T}}$. Let r_j be the root of C_j, i.e., $r_j = \min_{x \in C_j} x$, and let d_j be out-degree of r_j. We assume that the components are arranged in ascending order of the root, i.e., $r_1 < \cdots < r_t$. Let $\mathcal{T}_j$ be the family of coalitions $T \in \mathcal{T}$ in which all members have types in C_j. We write $T_{\in C}$ to denote $T_{\in C} = \{a \in T \mid x_a \in C\}$ for a coalition T and a set of types C; then $\mathcal{T}_j = \{T \in \mathcal{T} \mid T = T_{\in C_j}\}$. By definition of $G^{\mathcal{T}}$, $\{\mathcal{T}_1, \ldots, \mathcal{T}_t\}$ is a partition of $\mathcal{T}$.

By star-tree property of C_j, vertices $C_j \setminus \{r_j\}$ forms d_j paths in $G^{\mathcal{T}}$. Let C_j^ℓ ($\ell = 1, \ldots, d_j$) be the vertex sets of such paths. Then by Split lemma, we have the following three conditions:

$$\text{each } T \in \mathcal{T}_j \text{ satisfies either } \emptyset \neq T \subseteq A_{=r_j} \text{ or } A_{\in C_j^\ell} \subsetneq T \subseteq A_{\in C_j^\ell} \cup A_{=r_j} \text{ for some } \ell, \tag{3}$$

$$|T| = |T'| \text{ holds for any } T, T' \in \mathcal{T}_j, \text{ and} \tag{4}$$

$$|A_{\in C_j^\ell}| \neq |A_{\in C_j^h}| \text{ for any distinct } \ell, h \in [d_j]. \tag{5}$$

By (3), some agents of type r_j form a coalition T or some agents of type r_j together with the agents of types in C_j^ℓ form a coalition. It follows from (4) that each coalition in $\mathcal{T}_j$ has the same size λ_j. Let us call $\lambda^{\mathcal{T}} = (\lambda_1^{\mathcal{T}}, \ldots, \lambda_t^{\mathcal{T}})$ the *size vector* of $\mathcal{T}$. In summary, we have the following result as stated in Lemma 6, where isomorphism $\simeq$ of two allocations $\mathcal{T} = (T_1, \ldots, T_\alpha)$ and $\mathcal{T}' = (T'_1, \ldots, T'_\beta)$ is defined as follows: for two coalitions T and T', we write $T \simeq T'$ to mean that T and T' contains the same number of agents for each type, i.e., $|T_{=y}| = |T'_{=y}|$ for all $y \in V$; for two allocations $\mathcal{T}$ and $\mathcal{T}'$, we write $\mathcal{T} \simeq \mathcal{T}'$ if $|\mathcal{T}| = |\mathcal{T}'|$ and there exists a permutation $\sigma\colon [\alpha] \to [\alpha]$ such that $T_i \simeq T'_{\sigma(i)}$ for all $i \in [\alpha]$.

Lemma 6 *Suppose that an allocation $\mathcal{T}$ satisfies the conditions in Lemma 4. Then $G = G^{\mathcal{T}}$ and $\lambda = \lambda^{\mathcal{T}}$ satisfy the following conditions:*

$$G \text{ is a star-forest with (5) for any } j \text{ in } [t], \text{ and} \tag{6}$$

$$\text{for any } j \text{ in } [t], \lambda_j \text{ is a divisor of } |A_{\in C_j}| \text{ such that } \max_{\ell \in [d_j]} |A_{\in C_j^\ell}| < \lambda_j \leq |A_{\in C_j}|/d_j. \tag{7}$$

Conversely, if G and λ satisfy the conditions above, then there exists a unique allocation $\mathcal{T}$ (up to isomorphism) satisfying $G^{\mathcal{T}} = G$, $\lambda^{\mathcal{T}} = \lambda$, and the conditions in Lemma 4.

PROOF: Suppose that an allocation $\mathcal{T}$ satisfies the conditions in Lemma 4. It is not difficult to see that (6) follows from the discussion above and (5), and (7) follows from (3) and (4). Conversely, if G and λ satisfy (6) and (7), then we can construct a unique allocation $\mathcal{T}$ up to isomorphism that satisfies (3), (4), and (5). Thus $\mathcal{T}$ satisfies the conditions in Lemma 4. □

Notably, a unique allocation $\mathcal{T}$ in the converse statement can be computed in polynomial time if G and λ are given. Thus, a naive approach to find an envy-free feasible allocation is to enumerate all possible G and λ, and then

check if they provide a envy-free feasible allocation. Note that the number of star-forests is at most p^p, because the in-degree of every node is at most one. However, we may have $n^{\Omega(p)}$ many candidates of λ, even if a star-forest G is fixed in advance. To mitigate this difficulty, we show that for a given star-forest G, the size vectors λ such that G and λ provide envy-free feasible allocations form a semi-lattice. More precisely, for a star-forest G, let Λ_G denote the set of size vectors λ such that G and λ provide envy-free feasible allocations. Then we have the following structural property of Λ_G

Lemma 7 *For any star-forest G, Λ_G is an upper semi-lattice with respect to the componentwise max operation $\vee$, i.e., $\lambda, \lambda' \in \Lambda_G$ implies $\lambda \vee \lambda' \in \Lambda_G$*

We here remark that Λ_G may be empty. Based on this semi-lattice structure, we construct a polynomial time algorithm to compute an envy-free feasible allocation consistent with a given star-forest G. Specifically, for a given star-forest G, our algorithm computes the maximum vector in Λ_G or concludes that $\Lambda_G = \emptyset$, where the maximum vector exists due to the semi-lattice property of Λ_G. The lemma below ensures that it is possible in polynomial time. Since there exists at most p^p many star-forests, this implies an FPT algorithm (with respect to p) for computing an envy-free feasible allocation.

Lemma 8 *For a star-forest G, let $\Lambda = \prod_{j\in[t]} \Lambda_j$ be a non-empty set such that $\Lambda \supseteq \Lambda_G$. If the maximum vector $\lambda = (\max \Lambda_j)_{j\in[t]}$ does not belong to Λ_G, then an index $\ell \in [t]$ with $(\Lambda_\ell - \max \Lambda_\ell) \times \prod_{j\in[t]-\ell} \Lambda_j \supseteq \Lambda_G$ can be computed in polynomial time.*

Note that an index ℓ in the lemma must exist again by the semi-lattice property of Λ_G. Let $\Lambda = \prod_{j\in[t]} \Lambda_j$ denote a set of candidate size vectors. By Lemma 6, we have $\Lambda_G \subseteq \prod_{j\in[t]} \left[|A_{\in C_j}|\right]$. Our algorithm initializes Λ by $\Lambda = \prod_{j\in[t]} \left[|A_{\in C_j}|\right]$, and iteratively checks if $\Lambda = \emptyset$ or the maximum vector in Λ provides an envy-free allocation; If not, it updates Λ by utilizing indices ℓ in Lemma 8, where the formal description of the algorithm can be found in Algorithm 1.

Below, we show the following lemma, which is stronger than both Lemmas 7 and 8.

Lemma 9 *Let G be a star-forest, and let $\Lambda = \prod_{j\in[t]} \Lambda_j$ be a non-empty set in $\mathbb{Z}^t_{>0}$. If the maximum vector $\lambda = (\max \Lambda_j)_{j\in[t]}$ does not belong to Λ_G, then there exists an index $\ell \in [t]$ such that*

$$\left((\Lambda_\ell - \max \Lambda_\ell) \times \prod_{j\in[t]-\ell} \Lambda_j\right) \cap \Lambda_G = \Lambda \cap \Lambda_G. \tag{8}$$

In addition, such an index ℓ can be computed in polynomial time.

We note that Lemma 9 implies the semi-lattice property of Λ_G. To see this, suppose that Λ_G is not a semi-lattice, i.e., there exists two size vectors $\lambda, \lambda' \in \Lambda_G$ such that $\lambda \vee \lambda' \notin \Lambda_G$. Then we define Λ by $\Lambda_i = [(\lambda \vee \lambda')_i]$ for $i \in [t]$. By definition, $\lambda, \lambda' \in \Lambda$ and $\lambda \vee \lambda'$ is the maximum vector in Λ such that $\lambda \vee \lambda' \notin \Lambda_G$. However, no index ℓ satisfies (8), since the right-hand side of (8) contains both λ, λ', while the left-hand side of (8) contains at most one of them. Furthermore, if a set Λ in Lemma 9 is chosen in such a way that $\Lambda \supseteq \Lambda_G$, we obtain Lemma 8.

In order to show Lemma 9, let us consider the feasibility and monotonicity of allocations in addition to split property.

Lemma 10 *An allocation $\mathcal{T}$ is feasible and satisfies the conditions in Lemmas 3 and 4. Then $\lambda = \lambda^{\mathcal{T}}$ satisfy the following conditions.*

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_t \tag{9}$$

$$\textstyle\sum_{j\in[t]} |A_{\in C_j}|/\lambda_j \leq k, \text{ and} \tag{10}$$

$$\lambda_j \leq q_{\eta(j)} \text{ for all } j \in [t]. \tag{11}$$

where $\eta(j) = \sum_{r\leq j} |A_{\in C_r}|/\lambda_r$. Conversely, if G and λ satisfy (6), (7), (9), (10), and (11), then there exists a unique feasible allocation $\mathcal{T}$ (up to isomorphism) satisfying $G^{\mathcal{T}} = G$, $\lambda^{\mathcal{T}} = \lambda$, and the conditions in Lemmas 3 and 4.

PROOF: Suppose that $\mathcal{T}$ is a feasible allocation satisfying the conditions in Lemmas 3 and 4. By our assumption $r_1 < \dots r_t$, (1) implies (9). Note that the feasibility of $\mathcal{T}$ is equivalent to two conditions (i) $|\mathcal{T}| \leq k$ and (ii) capacity condition (i.e., $|T_i| \leq q_i$). Since $\mathcal{T}_j$ uses $|A_{\in C_j}|/\lambda_j$ many taxis, (i) is equivalent to (10). By (9) and the assumption $q_1 \geq \dots q_k$, in order to check capacity condition, it is enough to consider an allocation $\mathcal{T} = (T_1, \dots, T_\alpha)$ in such a way that $\mathcal{T}_1$ is assigned to the first $\eta(1)$ taxis, $\mathcal{T}_2$ is assigned to next $\eta(2) - \eta(1)$ taxis, and so on. More precisely, we have

$$\mathcal{T}_j = \{T_{\eta(j-1)+1}, \dots, T_{\eta(j)}\} \text{ for all } j \in [t],$$

where $\eta(0)$ is defined by 0. Thus the capacity condition implies (11). Conversely, if G and λ satisfy (6) and (7), then then by Lemma 3, there exists a unique allocation $\mathcal{T}$ (up to isomorphism) satisfying $G^{\mathcal{T}} = G$, $\lambda^{\mathcal{T}} = \lambda$, and the conditions in Lemma 4. Moreover, since (9), (10), and (11) hold for λ, $\mathcal{T}$ is feasible and the conditions in Lemma 3 are satisfied. □

Now, we prove Lemma 9.

Proof of Lemma 9: Let us separately consider the cases in which G and $\lambda = (\max \Lambda_j)_{j \in [t]}$ violate (6), (7), (9), (10), (11), and envy-freeness of the allocation provided by them.

- If (6) or (10) is violated, then by Lemmas 6 and 10, we have $\Lambda_G = \emptyset$. This implies that any index ℓ satisfies (8). Thus it is polynomially computable.
- If (7) is violated for an index j, then $\ell = j$ satisfies (8). Thus it is polynomially computable.
- If (9) is violated for an index j, i.e., $\lambda_{j-1} < \lambda_j$, then $\ell = j$ satisfies (8). Thus it is polynomially computable.
- If (11) is violated for an index j, i.e., $\lambda_j > q_{\eta(j)}$, then we claim that $\ell = j$ satisfies (8), which completes the proof of this case, since such an ℓ can be computed in polynomial time. Let λ' be a size vector in Λ such that $\lambda'_\ell = \lambda_\ell$, and let $\eta'(h) = \sum_{r \leq h} |A_{\in C_r}|/\lambda'_r$ for $h \in [t]$. Since $\lambda' \leq \lambda$ and $\lambda'_\ell = \lambda_\ell$, we have $\lambda'_\ell = \lambda_\ell > \eta(\ell) \geq \eta'(\ell)$, which implies the claim.
- Suppose that G and λ fulfill all the conditions above, i.e., G and λ provide a feasible allocation $\mathcal{T}$ that satisfies the conditions in Lemmas 3 and 4. Let further assume that $a \in T\,(\in \mathcal{T}_h)$ envies $a' \in T'\,(\in \mathcal{T}_j)$ for some $j, h \in [t]$. If $j = h$, then it is clear that $\ell = j\,(= h)$ satisfies (8). On the other hand, if $j \neq h$, Let λ' be a size vector in Λ such that $\lambda'_\ell = \lambda_\ell$ and satisfies (7), (9), (10), and (11). Then a still envies a' in the allocation provided by G and λ'. Thus $\ell = j$ again satisfies (8). Since envy-freeness can be checked in polynomial time, this completes the proof. □

Theorem 11 *We can check the existence of an envy-free feasible allocation, and find one if it exists in FPT with respect to the number p of types of agents.*

PROOF: We show that Algorithm 1 can check the existence of an envy-free feasible allocation and find one if it exists in FPT time. The correctness follows from Lemmas 6, 10, and 8. To analyze the running time, observe that the number of iterations of the while loop is at most n because $\sum_{j \in [t]} |\Lambda_j| = n$ at the beginning of the loop and it is decremented by at least one in each iteration. The running time of each iteration of the while loop is $O(n^3)$ because we can check the existence of envy in $O(n^3)$ time. Thus, the total running time of the algorithm is $O(p^p \cdot n^4)$, which is FPT with respect to p. □

Acknowledgement

This work was partially supported by the joint project of Kyoto University and Toyota Motor Corporation, titled "Advanced Mathematical Science for Mobility Society," JST PRESTO Grant Numbers JPMJPR2122 and JPMJPR20C1, and JSPS KAKENHI Grant Numbers JP19K22841, JP20H00609, JP20H05967, and JP22H00513.

Algorithm 1: FPT w.r.t. the number of destination types

foreach *star-forest* G **do**
  Let $\Lambda = \prod_{j\in[t]} \left[|A_{\in C_j}|\right]$;
  while $\Lambda \neq \emptyset$ **do**
    Let $\lambda = (\max \Lambda_j)_{j\in[t]}$;
    if *(6) or (10) is violated* **then**
      Set $\Lambda \leftarrow \emptyset$;
    else if *(7), (9), or (11) is violated for an index* j **then**
      Set $\Lambda_j \leftarrow \Lambda_j - \max \Lambda_j$;
    else if *an allocation* $\mathcal{T}$ *provided by* G *and* λ *is not envy-free, i.e., an agent in some coalition in* $\mathcal{T}_j$ *is envied* **then**
      Set $\Lambda_j \leftarrow \Lambda_j - \max \Lambda_j$;
    else
      return an allocation $\mathcal{T}$ provided by G and λ;
return "No envy-free feasible allocation";

References

[1] Javier Alonso-Mora, Samitha Samaranayake, Alex Wallar, Emilio Frazzoli, and Daniela Rus. On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. *Proceedings of the National Academy of Sciences*, 114(3):462–467, 2017.

[2] Yuki Amano, Ayumi Igarashi, Yasushi Kawase, Kazuhisa Makino, and Hirotaka Ono. Fair ride allocation on a line. In Panagiotis Kanellopoulos, Maria Kyropoulou, and Alexandros Voudouris, editors, *Algorithmic Game Theory*, pages 421–435, Cham, 2022. Springer International Publishing.

[3] Itai Ashlagi, Maximilien Burq, Chinmoy Dutta, Patrick Jaillet, Amin Saberi, and Chris Sholley. Edge weighted online windowed matching. In *Proceedings of the 2019 ACM Conference on Economics and Computation*, EC '19, page 729–742, New York, NY, USA, 2019. Association for Computing Machinery.

[4] H. Aziz and R. Savani. Hedonic games. In F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A.D. Procaccia, editors, *Handbook of Computational Social Choice*, chapter 15. Cambridge University Press, 2016.

[5] Siddhartha Banerjee, Yash Kanoria, and Pengyu Qian. State dependent control of closed queueing networks. In *Abstracts of the 2018 ACM International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '18, pages 2–4, New York, NY, USA, 2018. Association for Computing Machinery.

[6] Nathanaël Barrot and Makoto Yokoo. Stable and envy-free partitions in hedonic games. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, IJCAI'19, page 67–73. AAAI Press, 2019.

[7] Hans L. Bodlaender, Tesshu Hanaka, Lars Jaffke, Hirotaka Ono, Yota Otachi, and Tom C. van der Zanden. Hedonic seat arrangement problems. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '20, page 1777–1779, Richland, SC, 2020. International Foundation for Autonomous Agents and Multiagent Systems.

[8] Anna Bogomolnaia and Matthew O. Jackson. The stability of hedonic coalition structures. *Games and Economic Behavior*, 38(2):201–230, 2002.

[9] Georgios Chalkiadakis, Edith Elkind, and Michael Wooldridge. Computational aspects of cooperative game theory. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 5(6):1–168, 2011.

[10] Youngsub Chun and Boram Park. The airport problem with capacity constraints. *Review of Economic Design*, 20(3):237–253, 2016.

[11] YOUNGSUB CHUN, CHENG-CHENG HU, and CHUN-HSIEN YEH. A strategic implementation of the shapley value for the nested cost-sharing problem. *Journal of Public Economic Theory*, 19(1):219–233, 2017.

[12] Duncan K. Foley. Resource allocation and the public sector. *Yale Economic Essays*, 7:45–98, 1967.

[13] Jonathan Goldman and Ariel D. Procaccia. Spliddit: Unleashing fair division algorithms. *SIGecom Exchange*, 13(2):41–46, 2015.

[14] Maria Kyropoulou, Warut Suksompong, and Alexandros A. Voudouris. Almost envy-freeness in group resource allocation. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, IJCAI'19, page 400–406. AAAI Press, 2019.

[15] S. C. Littlechild and G. Owen. A simple expression for the shapley value in a special case. *Management Science*, 20(3):370–372, 1973.

[16] Dov Monderer and Lloyd S. Shapley. Potential games. *Games and Economic Behavior*, 14(1):124–143, 1996.

[17] Marco Pavone, Stephen L Smith, Emilio Frazzoli, and Daniela Rus. Robotic load balancing for mobility-on-demand systems. *The International Journal of Robotics Research*, 31(7):839–854, 2012.

[18] Dominik Peters and Edith Elkind. Simple causes of complexity in hedonic games. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, page 617–623. AAAI Press, 2015.

[19] Robert W. Rosenthal. A class of games possessing pure-strategy nash equilibria. *International Journal of Game Theory*, 2(1):65–67, 1973.

[20] Paolo Santi, Giovanni Resta, Michael Szell, Stanislav Sobolevsky, Steven H. Strogatz, and Carlo Ratti. Quantifying the benefits of vehicle pooling with shareability networks. *Proceedings of the National Academy of Sciences*, 111(37):13290–13294, 2014.

[21] Erel Segal-Halevi and Shmuel Nitzan. Fair cake-cutting among families. *Social Choice and Welfare*, 53(4): 709–740, 2019.

[22] L. S. Shapley. A value for n-person games. In Harold William Kuhn and Albert William Tucker, editors, *Contributions to the Theory of Games II*, pages 307–317. Princeton University Press, Princeton, 1953.

[23] William Thomson. Cost allocation and airport problems. RCER Working Papers 537, University of Rochester - Center for Economic Research (RCER), 2007.

[24] Rick Zhang and Marco Pavone. Control of robotic mobility-on-demand systems: A queueing-theoretical perspective. *The International Journal of Robotics Research*, 35(1–3):186–203, 2016.

Polynomial-Time Algorithm for the Regional SRLG-disjoint Paths Problem

Balázs Vass [1]

Department of Telecommunication and Media Informatics
University of Technology and Economics (BME)
Budapest, Hungary
balazs.vass@tmit.bme.hu

Erika Bérczi-Kovács [2]

Alfréd Rényi Institute of Mathematics and
ELKH-ELTE Egerváry Research Group on Combinatorial Optimization
Budapest, Hungary
erika.berczi-kovacs@ttk.elte.hu

Ábel Barabás

Department of Operations Research
Eötvös Loránd University
Budapest, Hungary
barabasabel@gmail.com

Zsombor László Hajdú [1]

Department of Telecommunication and Media Informatics
University of Technology and Economics (BME)
Budapest, Hungary
hajdu@tmit.bme.hu

János Tapolcai [1]

Department of Telecommunication and Media Informatics
University of Technology and Economics (BME)
Budapest, Hungary
tapolcai@tmit.bme.hu

Abstract: The current best practice in survivable routing is to compute link or node disjoint paths in the network topology graph. It can protect single-point failures; however, several failure events may cause the interruption of multiple network elements. The set of network elements subject to potential failure events is called Shared Risk Link Group (SRLG), identified during network planning. Unfortunately, for any given list of SRLGs, finding two paths that can survive a single SRLG failure is NP-Complete. In this paper, we provide a polynomial-time SRLG-disjoint routing algorithm for planar network topologies and a large set of SRLGs. Namely, we focus on regional failures, where the failed network elements must not be far from each other. We use a flexible definition of regional failure, where the only restrictions are that i) the topology is a planar graph, ii) each SRLG forms a set of connected edges in the dual of the planar graph, and iii) for each node v, the links incident to v are part of an SRLG. The proposed algorithm is based on a max-min theorem.

Keywords: planar graphs, disjoint paths, regional failures

1 Introduction

Disjoint path computation is the essence of any strategy for networks to survive failures. The current best practice is to utilize network flow algorithms, such as Suurballe's algorithm [14], to efficiently compute

[1] Balázs Vass, Zsombor László Hajdú, and János Tapolcai are also with MTA-BME Future Internet Research Group and ELKH-BME Information Systems Research Group. {,hajdu,tapolcai}@tmit.bme.hu.

[2] Also with Department of Operations Research, Eötvös Loránd University (ELTE), Budapest, Hungary.

link or node disjoint paths in the network topology graph. However, several papers studied [11, 3, 4] that the networks have severe outages when almost every equipment in a vast physical region gets down as a result of a disaster, such as earthquakes, hurricanes, tsunamis, tornadoes, etc. These types of failures are called *regional failures*, which are simultaneous failures of nodes/links located in specific geographic areas. The set of network links subject to potential failure events is called Shared Risk Link Group (SRLG), identified during network planning [2].

Unfortunately, for a given list of SRLGs and topology graph, finding two paths that can survive a single SRLG failure is NP-Complete in general [5]. The proof is a reduction from 3SAT where each SRLG corresponds to a clause in the formula. Roughly speaking, a very artificial topology graph and SRLG settings are needed to show the high computational complexity of the problem, and many believe SRLG-disjoint routing is a well-solvable problem in practice. For example, Kobayashi-Otsuki provided [6] a routing algorithm for circular disk failures of fixed radius in a planar graph topology where the links are straight lines. Circular disk failures of the fixed radius are the most well studied regional failure model, see [11, 15]. Naturally arises the question: *Is there another set of regional SRLGs for which the SRLG-disjoint routing problem is solvable in polynomial time? Can we define a simple and general property of the regional SRLGs to have efficient routing algorithms?* The paper provides a positive and surprisingly simple answer as follows.

This study assumes the network topology is a planar graph. In backbone optical networks, it is rare that cables cross each other without having an optical cross-connect at the intersection. Planarity is an natural assumption to have a polynomial-time algorithm for an otherwise NP-hard problem.

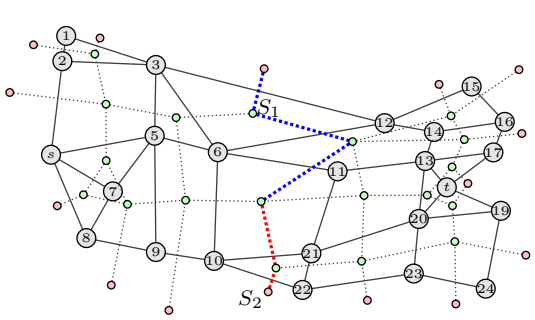
Apart from that, we adopt a very general model, here we may consider the network is somehow embedded on the Earth's surface, the links are curved lines between the endpoints, and an SRLG is resulting from a connected disaster area. We assume the list of SRLGs is defined in the service level agreement (SLA) [13] at network planning. The list of SRLGs typically involves physically close network nodes and parallel links, might be computed by any regional failure model [17], or based on historical data of natural disasters, such as earthquakes [18], tornadoes, tsunamis, electromagnetic pulse (EMP) attacks, etc [9] .

Furthermore, the proposed routing algorithms do not even require knowing the geometry of the network, such as node coordinates and route of the cables. It is necessary because the router's routing engine cannot have such geographic information. The exact location of the network equipment is sensitive information for military and economic reasons, which will never be widely distributed on the internet. Note that, often, the network operators do not have any information about the route of the links or the physical coordinates of the intermediate routing nodes because the links are hired as a service from an independent company [1], called the Physical Infrastructure Provider. After all, information on the routes of the links is not part of any network protocol so far. So the key idea of our approach is that knowing the dual of the planar topology graph is sufficient for the routing computations, and also we will define only combinatorial properties that the SRLGs must meet.

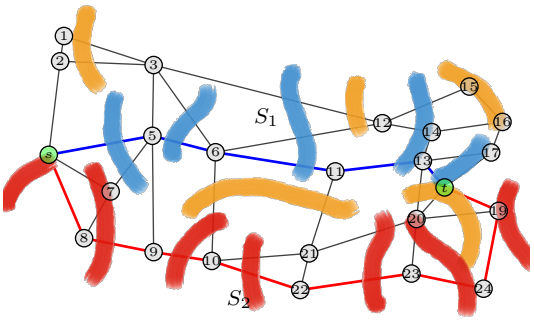
Fig. 1a shows such an example input: a planar topology graph with its dual graph. The nodes of the dual graph are the faces, and there are edges between the adjacent faces. Thus, each link e of the topology graph has a corresponding dual-edge, whose endpoints are the dual vertices corresponding to the faces on either side of e. Therefore, an SRLG as a set of links can be mapped to a set of dual-edges.

To mitigate the above problem, we assume the routing engine knows the dual graph of the planar network topology with the mapping between the links and dual-edges. The only assumption we have for SRLGs, that the corresponding dual-edges are connected. Note that it is a very loose restriction and covers all SRLGs that correspond to a connected disaster area. Here the disaster area is the geographic (connected) region in which the network elements are subject to fail simultaneously. A regional failure disconnects a link if it contains at least one (possibly end node) point of that link. For example the SRLGs S_1 and S_2 shown on Fig. 1b correspond to the dual-edges colored red and blue on Fig. 1a that are connected in the dual graph.

We provide a broad definition of 'regional SRLG,' where the regional SRLG-disjoint routing can be efficiently solved. For this, we define a pure combinatorial routing problem input, which contains a planar network topology and the corresponding dual graph. We show that this input is sufficient for efficient



(a) The US network topology graph (G) with its dual (G^*). The dual nodes are drawn with small green, and the outer region is the red dual node, split on the illustration into multiple nodes. The dual-edges are drawn with dotted lines and intersect the corresponding network links. The duals of two SRLGs, S_1 and S_2, are highlighted.



(b) The regional SRLGs ($\mathcal{S}_{region}$) are hand drawn with brush, and colored with the same color of the path traversed by, otherwise orange. The full list of SRLGs also include every single link or node failures as well. Two SRLG-disjoint paths between the source (s) and the target (t) node are drawn with red and blue links.

Input: a planar graph $G = (V, E)$, for every node the cyclic order of incident links in a planar drawing, two distinct nodes $s, t \in V$, and a set $\mathcal{S} \subseteq 2^{|E|}$ of dual-connected SRLGs with $\mathcal{S}_V \subseteq \mathcal{S}$.

Maximum Regional SRLG-disjoint Paths Problem (MRSDP): **Find:** maximum cardinality set of pairwise $\mathcal{S}$-disjoint s-t paths.

Figure 1: Illustration of the problem. Dual-edges corresponding to a regional SRLG are connected in the dual graph, for example, SRLG S_1 on (b) is mapped to blue dual-edges on (a). Note that SRLGs S_1 and S_2 forms an s-t cut, thus, there can be at most two SRLG-disjoint s-t paths.

routing computations, and no other information on the geometry of the physical topology is needed. We have a very flexible definition of regional failure, where we assume the SRLGs mapped to the dual-edges of the planar graph are connected. We provide an efficient polynomial-time SRLG-disjoint routing algorithm for the regional SRLG model defined above and planar network topology.

The paper is organized as follows. In Subsec. 1.1 we summarize previous theoretical work on the topic. Sec. 2 provides the problem formulation, the main results and a simple upper bound on the number of SRLG-disjoint paths. Sec. 3 describes the proposed algorithm.

For an extended version of this paper see [16].

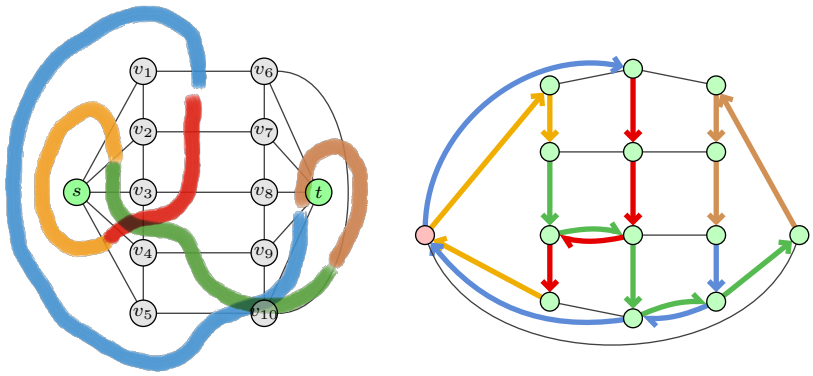
1.1 Theoretical preludes

Papers [8] and [7] provided polynomial algorithms and min-max theorems to find a maximal number of interiorly d-hop disjoint paths (i.e., no walk of length d is connecting any pair of these paths) in planar graphs, for $d = 1$, and $d \geq 1$, respectively. The condition of interiorly d-hop disjointness can be rephrased as interiorly SRLG-disjointness for a special class of primal-connected SRLGs.

Based on the former, and motivated by [10], [6] and [12] designed a tight min-max theorem and faster polynomial algorithms for finding a maximal number of circular disk-disjoint paths in geometric graphs without link crossings. The disk-disjointness can be rephrased as SRLG-disjointness for a special class of dual-connected SRLGs.

2 Problem formulation and main result

Let $G = (V, E)$ be a planar network topology graph with a *node* set V, a *link* set E, and two distinct nodes $s, t \in V$. We do not know any geometric embedding of G, instead we only know the order of incident links at every node in the embedding. Note that from this information the dual graph $G^* = (V^*, E^*)$



(a) The network topology and the SRLGs ($\mathcal{S}_{region}$) are drawn with brush of unique color.

(b) The dual graph with a closed dual walk C such that $l(C) = 5$, $w(C) = 3$, and hence $^{l(C)}/_{w(C)} < 2$.

Figure 2: A graph, where the MIN-CUT $= 3$, but there is no two SRLG-disjoint paths between s and t, meaning MAX-FLOW $=$ MIN-CUT $- 2$.

can be easily calculated. When it does not confuse, we identify the faces of G with their dual nodes in $G^* = (V^*, E^*)$. In other words $G^* = (V^*, E^*)$ is composed of a *face* set V^* and a *dual-edge* set E^*, see Fig. 1a. In what follows, a link is sometimes called an edge.

Let $\mathcal{S}_{region} \subseteq 2^{|E|}$ be a set of link sets representing a set of *regional SRLGs*. We assume the set of SRLGs also contains all the single node failures, which ensures the obtained SRLG-disjoint paths to be node-disjoint. Let E_v denote the set of links in G incident to a node v and let $\mathcal{S}_V$ represent the set of SRLGs modeling the node failures, i.e.,

$$\mathcal{S}_V = \{E_v | v \in V \setminus \{s, t\}\} \ .$$

Let $\mathcal{S}$ denote the set of all SRLGs: $\mathcal{S} = \mathcal{S}_{region} \cup \mathcal{S}_V$. Let ρ denote the maximum size of a regional SRLG: $\rho := \max\{|S| | S \in \mathcal{S}\}$, and let μ denote the maximum number of SRLGs that contain the same edge: $\mu = \max\{|T| : T \subset \mathcal{S}, |\cap_{S \in T} S| > 0\}$. We say that two paths are **($\mathcal{S}$-)disjoint** or SRLG disjoint if there is no SRLG $S \in \mathcal{S}$ intersecting both of them[1]. We may omit $\mathcal{S}$ from the notation when the SRLG set is clear from the context.

Formally, for a link set $X \subseteq E$, let X^* be the set of duals of links of X. For an SRLG $S \in \mathcal{S}$, let $V^*(S^*) := \{f \in V^* | \text{there is a dual-edge } \{f, f'\} \in S^* \text{ for some } f'\}$. We denote by d the maximal diameter of the dual of an SRLG: $d := \max\{\, diam(S^*) | S \in \mathcal{S}_{region}\}$, where $diam(S^*) = \max_{f,f' \in V^*(S^*)} \min\{$ edge lengths of f-f' paths in $S^*\}$. We call a set of links $S \subseteq E$ **dual connected**, if the edge-induced subgraph of S^* is connected in G^*. For example, each $E_v \in \mathcal{S}_V$ is clearly dual connected. We demand $\mathcal{S}$ to fulfill the following property:

Property 1 *Each set $S \in \mathcal{S}$ is dual connected.*

Recall we have a second property:

Property 2 *All node failures are listed apart from s and t ($\mathcal{S}_V \subseteq \mathcal{S}$).*

Our main goal in this paper is to find the maximum number of $\mathcal{S}$-disjoint s-t paths in planar graphs and SRLG sets with Properties 1, 2, which we call **Maximum Regional SRLG-disjoint Paths Problem (MRSDP)**. See Figure 1 for the exact problem definition. Let MAX-FLOW denote the optimal value of the problem. First, we give a trivial upper bound on MAX-FLOW using the analogy of max-flow min-cut

[1] In the related literature, 'disjointness' is sometimes called 'separatedness'.

theorems for network flows. A set of SRLGs from $\mathcal{S}$ that disconnect s from t is called an **SRLG cut** in this paper, see SRLG S_1 and S_2 on Fig. 1b as an illustration. It is easy to see that the size of an SRLG cut is an upper bound for MAX-FLOW, because two disjoint paths cannot traverse any of these SRLGs simultaneously by definition. Let MIN-CUT denote the minimum size of an SRLG cut. Fig. 2a shows an example graph where the MAX-FLOW = 1, while MIN-CUT = 3. The gap between the MAX-FLOW and MIN-CUT is at most 2 (for a proof see [16]).

Theorem 1 *For any instance of the MRSDP problem and its corresponding MAX-FLOW and MIN-CUT values we have*

$$\textit{MAX-FLOW} \leq \textit{MIN-CUT} \leq \textit{MAX-FLOW} + 2.$$

Although MIN-CUT does not give a sharp upper bound for MAX-FLOW, a min-max characterization can be given, which is one of the main results of this paper. In order to state this sharp upper bound, we need a more complex structure than a cut. Here we intuitively present the necessary notions, which are precisely defined in Sec. 2.1. A **walk** is a finite sequence of edges which joins a sequence of vertices. For a closed walk C in the dual graph G^*, the length $l(C)$ is the minimum number of times one has to "switch SRLG" to go around C, while the winding number $w(C)$ of C is the number of times that C separates s and t. Our main result is the following.

Theorem 2 *For any instance of the MRSDP problem, we can find a maximum number of $k = $ MAX-FLOW SRLG disjoint paths in $O\left(n^2\mu\left(\log k + \rho \log d\right)\right)$ time, and we determine closed dual walk C in G^*, for which $\left\lfloor \frac{l(C)}{w(C)} \right\rfloor = k$. For MAX-FLOW ≥ 2 we also have*

$$\textit{MAX-FLOW} = \min\left\{ \left\lfloor \frac{l(C)}{w(C)} \right\rfloor \middle| C \text{ closed dual walk, } w(C) \geq 1 \right\}.$$

2.1 Upper Bounds on the Number of Maximum Regional SRLG-disjoint Paths

In this section, we will provide another upper bound for MAX-FLOW by generalizing the approach of [6]. This upper bound will turn out to be tight (cf. Thm. 2). Let C be a closed walk in G^*. We define the **winding number** $w(C)$ of C as the number of times that C separates s and t. More precisely, let us fix an s-t path P in G, and consider the edges of P being oriented towards t. Let us consider a one-way orientation of the dual-edges of dual walk C. Let $w_1(C) = \{\#e_d \in C | e_d \text{ crosses an } e_p \in P \text{ from left to right}\}$. Similarly, $w_2(C) := \{\#e_d \in C | e_d \text{ crosses an } e_p \in P \text{ from right to left}\}$. Lastly, we define $w(C) := |w_1(C) - w_2(C)|$. E.g., the (colored) dual walk on Fig. 2b separates s and t three times. Note that if C is a closed walk, then $w(C)$ is indifferent to the choice of P and orientation of C.

Now we define $l(C)$ for a closed dual walk C. Let $C = \{C_1, ..., C_k\}$ be a partition of the dual-edges such that each C_i consists of consecutive edges of C, and there exists an SRLG $S_i \in \mathcal{S}$ such that S_i^* contains C_i. Let $l(C)$ be the minimal number for which there exists such a partition. For example, to cover the dual walk on Fig. 2b we need at least 5 SRLGs. We note that $l(C) \leq |V^*|$ will hold for the closed dual walks constructed in our proofs.

By using these notations, we can give an upper bound for MAX-FLOW as follows.

Lemma 3 *Consider an instance of the MRSDP problem. If MAX-FLOW ≥ 2, then*

$$\textit{MAX-FLOW} \leq \min\left\{ \left\lfloor \frac{l(C)}{w(C)} \right\rfloor \middle| C \text{ closed dual walk, } w(C) \geq 1 \right\}. \tag{1}$$

Proof: Suppose we have s-t paths $P_1, \ldots, P_{k\geq 2}$ that are pairwise disjoint and let $C = \{C_1, ..., C_{l(C)}\}$ be a closed dual-walk such that each subwalk C_j is contained by the dual of an SRLG $S_j \in \mathcal{S}$. We show that each P_i has to intersect at least $w(C)$ subwalks C_j. Observe that each C_j adds at most 1 to the value of $w(C)$: $w(C_j) := |w_1(C_j) - w_2(C_j)| \leq 1$, since paths P_i are vertex disjoint (by Property 2). Two disjoint paths cannot cross C at the same C_j, so we have $l(C) \geq k \cdot w(C)$. □

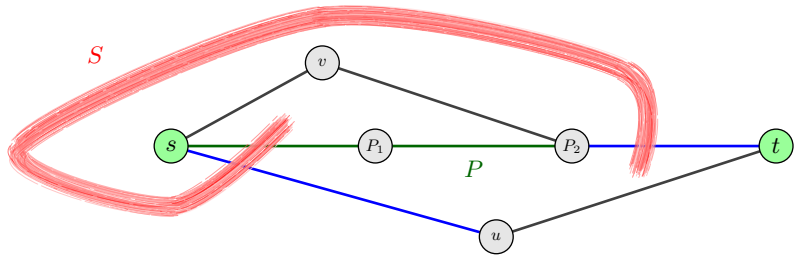


Figure 3: Clockwise part $\{su, P_2t\}$ of SRLG $S = \{su, sP_1, P_2t\}$ with respect to path $P = s, P_1, P_2, t$

3 Polynomial Time Algorithm to Find a Maximum Number of Regional SRLG-Disjoint Paths

In this section we show that Lemma 3 can be extended into exact min-max theorem for MAX-FLOW, and Eq. (1) holds with equality. If MAX-FLOW $= 1$, we give a closed dual walk C with $l(C)/w(C) < 2$. Our proof generalizes ideas in [6], which shows a geometric min-max theorem for the special case of the MRSDP problem, where the disaster regions are circular disks.

The algorithm has two main parts: the base case (see extended version in [16]) and the inductive part (3.1). The inductive part decides whether there exist k $\mathcal{S}$-disjoint paths, assuming that $k-1$ such paths are given as starting paths.

When searching for $k = 2$ $\mathcal{S}$-disjoint paths P_1 and P_2, for algorithmic reasons, the starting path needs to be 'clockwise far enough' from itself. We use the term clockwise $\mathcal{S}$-disjointness to capture the intuition precisely (see definition below). The goal of the base case is to decide whether there exists a path that is clockwise $\mathcal{S}$-disjoint from itself.

First we introduce the notion of crossings. We say two s-t paths P_1 and P_2 are **crossing** if, after contracting their common edges, there is a subpath P' contained by both paths such that the links entering/leaving P' in P_1 and P_2 are alternating according to their incidence to P_1 and P_2. We note that with this definition, two non-crossing paths may have common edges, intuitively, the only restriction for them is not to change their clockwise order along the way from s to t.

Now we turn to the definition of clockwise $\mathcal{S}$-disjointness. For an s-t path P in G and a directed dual path Q^* in G^* we say that Q^* is **clockwise to** P if it does not cross P from right to left, that is, $w_2(C^*) = 0$. For an s-t path P and an intersecting SRLG S we define $S_{\text{clw}}(P)$ the **clockwise part of** S **with respect to** P as the subset of those links in $S \setminus (S \cap P)$ for which the corresponding dual edge is reachable from $(S \cap P)^*$ on a path clockwise to P. (see Fig. 3).

For two s-t paths P_1 and P_2 without crossings, an ordered pair (P_1, P_2) is **clockwise ($\mathcal{S}$-)disjoint** if for any SRLG S in $\mathcal{S}$ intersecting P_1, $S_{\text{clw}}(P_1)$ does not intersect P_2. Obviously, paths P_1 and P_2 are disjoint exactly if both pairs (P_1, P_2) and (P_2, P_1) are clockwise disjoint.

3.1 Induction step

In what follows we show the equality in (1) for MAX-FLOW ≥ 2. First, we assume that for some $k \geq 2$ we have $k-1$ pairwise disjoint s-t paths $P_1, \ldots P_{k-1}$ (when $k = 2$ we assume that P_1 is clockwise disjoint from itself). We will give an algorithm for finding either k pairwise disjoint s-t paths or a closed dual walk C with $\lfloor l(C)/w(C) \rfloor = k-1$ (see Algorithm 1). Then applying the algorithm repeatedly for $k = 2, \ldots,$ MAX-FLOW, we get an inductive proof of the equality in Lemma 3. (How to find a starting path P_1 that is clockwise disjoint from itself is described in [16].)

We may assume that the first edges of $P_1, \ldots, P_{k-1}$ occur in this clockwise order at s. We continue this series of paths by generating new s-t paths $P_k, P_{k+1}, \ldots$. At each step, a new path P_l is generated and if $P_{l-k+1}, \ldots, P_l$ are pairwise disjoint, we stop. Otherwise we generate a new path again. If we do not find k pairwise disjoint paths after $|V^*| + 1$ path generations, then the algorithm stops and we can

Algorithm 1: Search for one more SRLG-disjoint path

Input: MRSDP problem input, $P_1, \ldots, P_{k-1}$ pairwise disjoint s-t paths if $k \geq 3$ or an s-t path P_1 that is clockwise disjoint from itself if $k = 2$.

Output: k pairwise disjoint s-t paths or a closed dual walk C in G^* with $\left\lfloor \frac{l(C)}{w(C)} \right\rfloor = k - 1$

1 $P_0 := P_{k-1}$
2 **for** $l = k, \ldots, k + |V^*|$ **do**
3 $P_l := P_{\text{nearest}}(P_{l-1}, P_{l-k})$ (see Alg. 2)
4 **if** P_l, P_{l-k+1} *are* $\mathcal{S}$*-disjoint* **then**
5 **return** $P_{l-k+1}, \ldots, P_{l-1}, P_l$
6 **return** *a closed dual walk* C *in* G^* *with* $\left\lfloor \frac{l(C)}{w(C)} \right\rfloor = k - 1$

determine a closed dual walk C with $\lfloor l(C)/w(C) \rfloor = k - 1$ (see Claim 5). Our algorithm is described in Algorithm 1.

When generating a new path P_l we use previous paths P_{l-1} and P_{l-k}. Intuitively, P_l is the path clockwise 'nearest' to P_{l-k} among those that are clockwise-disjoint from P_{l-1}.

Now we give the precise definition of 'nearness' by describing an ordering of the paths. The clockwise order of the links incident to a node v gives a cyclic ordering of those links. For a fixed link e incident to v this cyclic ordering induces a complete ordering $<_{v,e}$ of the links incident to v: for links e_1, e_2 incident to v we say that $e_1 <_{v,e} e_2$ if e_1 is earlier than e_2 in the clockwise order starting from e. Given an s-t path P, these orderings induce an ordering $<_P$ on the set of s-t paths the following way. Let P_1 and P_2 be s-t paths and let v denote the first node where they enter on the same link (say e) but continue on different links, say e_1 and e_2 (if $v = s$, let e be the first link of P). We say that $P_1 <_P P_2$ if $e_1 <_{v,e} e_2$.

Now we are ready to give a precise definition of P_l: it is an s-t path that is clockwise disjoint from P_{l-1}, does not cross P_{l-k} and within these constraints minimum with respect to $<_{P_{l-k}}$ (see Algorithm 2).

3.2 Computing the next nearest clockwise SRLG-disjoint path

In Algorithm 2 we have two non crossing paths Q_1, Q_2 as input such that Q_1 is clockwise disjoint from itself. We determine a path P that is clockwise-disjoint to Q_1, does not cross Q_2 and within these constraints minimum for $<_{Q_2}$. Note that by calling the algorithm with $Q_1 = P_{l-1}$ and $Q_2 = P_{l-k}$ we get the required path P_l in Algorithm 1.

Algorithm 2 uses DFS on a proper auxiliary graph G' and explores the nodes in clockwise order to find the optimal path. In order to avoid path P to cross Q_2, we modify G. We duplicate path Q_2 by 'cutting' it into two along its route, creating a left and a right copy of Q_2: instead of each internal node v on Q_2 we add two nodes v_{left} and v_{right} to G, and for each internal link $uv \in Q_2$ we add two links $u_{\text{left}}v_{\text{left}}$ and $u_{\text{right}}v_{\text{right}}$. For a link uv incident to a node $v \in Q_2$ but not on Q_2 we create the link $v_{\text{left}}u$ if uv is on the left side of Q_2 and we create $v_{\text{right}}u$ if the link is on the right side of Q_2. Similarly we add two copies of links of the form vu with v on Q_2 but u not on Q_2. The first and last links (say sv and ut) have two copies: sv_{left}, sv_{right} and $u_{\text{left}}t$, $u_{\text{right}}t$, respectively. Let G_{Q_2} denote the resulting graph. Note that G_{Q_2} is also planar, and there is a bijection between the s-t paths of G not crossing Q_2 and the s-t paths of G_{Q_2} (apart from Q_2, which has two copies in G_{Q_2}).

Clockwise separation to Q_1 can be guaranteed by deleting the clockwise part of all SRLG-s intersecting Q_1 (see line 3). If a link e to be deleted is in Q_2, we delete both the left and right copies of the link (see Fig. 4). The resulting graph is G'. Then an optimal path with respect to $<_{Q_2}$ can be easily determined by a DFS if we fix the order of node exploration according to the clockwise order of the links. Since Q_1 does not cross Q_2 and is clockwise disjoint from itself, Q_1 is in G'. Hence t is reachable from s in G' and the DFS finds an s-t path indeed.

Now we show by induction that the last $k - 1$ paths in the series behave similarly to the input paths.

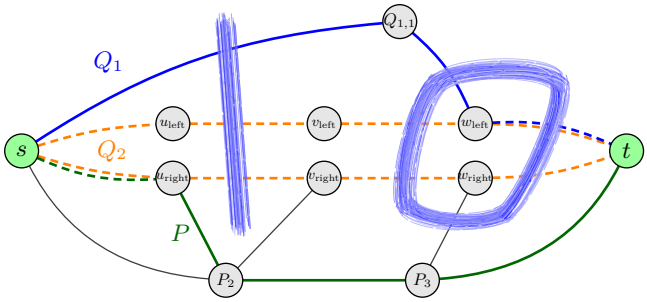


Figure 4: s-t path P that is minimum with respect to $<_{Q_2}$, clockwise-disjoint to Q_1 and does not cross Q_2. (Usually, we call Alg. 2 with $P = P_l$, $Q_1 = P_{l-1}$ and $Q_2 = P_{l-k}$)

Algorithm 2: Nearest clockwise SRLG-disjoint path

Input: Planar graph $G(V, E)$, SRLG set $\mathcal{S}$, non crossing s-t paths Q_1, Q_2, such that (Q_1, Q_1) is clockwise disjoint

Output: An s-t path P that is clockwise-disjoint to Q_1, does not cross Q_2, and is minimum with respect to $<_{Q_2}$

1 $G' := G_{Q_2}$
2 **for** $(v_1, v_2) \in E(Q_1)$ **do**
3 | **for** $S \in \mathcal{S} : (v_1, v_2) \in S$ **do**
| | $E' := E' \setminus S_{\text{clw}}(Q_1)$
4 DFS-TREE:= DFS tree on E' rooted at s, exploring nodes in clockwise order (see $<_{v,e}$).
Starting link of DFS: sq_{right}, where $sq \in Q_2$.
5 **return** *the s-t path in DFS-TREE*

Claim 4 *1. Paths $P_{l-k+2}, \ldots, P_l$ are pairwise $\mathcal{S}$-disjoint and in this clockwise order at s if $k \geq 3$.*

2. Path P_l is clockwise disjoint from itself if $k = 2$.

Proof: First, we prove part a). It is enough to show that the paths are in this clockwise order at s and that P_l and P_{l-k+2} are $\mathcal{S}$-disjoint. Since by induction P_{l-1} and P_{l-k+1} are $\mathcal{S}$-disjoint, they are also clockwise $\mathcal{S}$-disjoint and P_{l-k+1} does not cross P_{l-k}. We know that P_l is minimum with respect to $<_{P_{l-k}}$ among such paths, hence $P_l \leq_{P_{l-k}} P_{l-k+1}$, which shows the clockwise order of the paths. All we have to show is that P_l is clockwise $\mathcal{S}$-disjoint to P_{l-k+2}. Assume indirectly that there is an SRLG S such that there is a dual path $Q^* \subseteq S_{\text{clw}}(P_l)$ connecting dual edges e^*, f^* such that $e \in P_l, f \in P_{l-k+2}$. Since path P_{l-k+1} is between P_l and P_{l-k+2} in the clockwise order, this dual path would have a dual edge h^* such that $h \in P_{l-k+1}$ contradicting that P_{l-k+1} and P_{l-k+2} are clockwise $\mathcal{S}$-disjoint.

Now we similarly prove the second part of the claim. Assume indirectly that P_l is not clockwise disjoint and there are (not necessarily different) dual edges e^*, f^* such that there is a dual path connecting e^* to f^* in $S^*_{\text{clw}}(P_l)$. Then this dual path would have a dual edge h^* where $h \in P_{l-1}$, contradicting that P_{l-1} and P_l are clockwise disjoint. □

If we find pairwise disjoint paths $P_{l-k+1}, \ldots, P_{l-1}, P_l$ in line 5 of Algorithm 1, then we are done. In what follows, we give a procedure for finding a closed dual walk C with $l(C)/w(C) < k$ (line 6) when such paths do not appear while $l = k, k+1, \ldots, k+|V^*|$. Let $N := k + |V^*|$.

Claim 5 *For $i = N, \ldots, k$, we can compute links $e_i \in E$, faces $f_i \in V^*$, SRLGs $S_i \in \mathcal{S}$, and paths $C_i \subseteq S^*_i$ such that*

- *e_i is part of $P_i \setminus P_{i-k}$,*

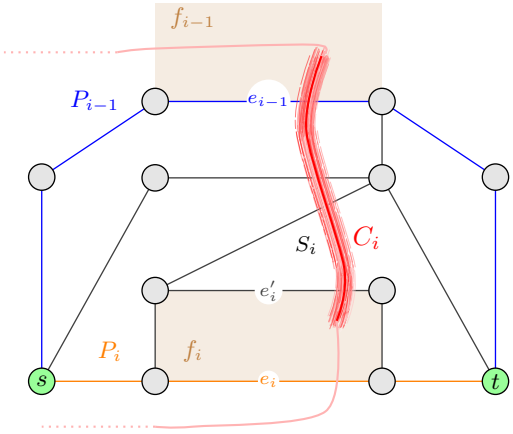


Figure 5: Illustration for Claim 5

- *f_i is the face left to e_i (as we walk on P_i from s to t)*
- *C_i is a dual path connecting f_{i-1} to f_i starting with e^*_{i-1} and then going in $S^*_{i\ clw}(P_{i-1})$.*

□

For $i = N, \dots, k$, let e_i, f_i, S_i, and C_i be as described in Claim 5. By pigeonhole principle, $f_i = f_j$ for some $k \leq i \leq j \leq N$. Let C be the closed dual walk yielding from the concatenation of $C_{i+1}, \dots, C_j$. We will show that C satisfies ${}^{l(C)}/_{w(C)} < k$, which is equivalent to $u := \lfloor {}^{(j-i)}/_{k} \rfloor < w(C)$, because $l(C) = j - i$. If $u = 0$, then the inequality is trivial. Otherwise, e_j is strictly to the right of P_{j-k} (by Claim 5).

By line 3 of Alg. 1, $P_{j-(l+1)k}$ is to the left of P_{j-lk} for all $l = 1, \dots, u$. Based on this, we can see that $C_{j-(l+1)k+1} \cdot \ldots \cdot C_{j-lk}$ makes at least one turn clockwise. Concentrating now on path P_N, we can see that we have an extra right-to-left crossing of the path at the last edge of C_{i+1}, that hitherto was not considered, which means $w(C_i \cdot \ldots \cdot C_j) = w(C) \geq u + 1$.

By the above procedure, we can find a closed dual walk C with ${}^{l(C)}/_{w(C)} < k$ in line 6 of Algorithm 1. Since the input of the Algorithm was a number of $k-1$ SRLG-disjoint paths, we also have $k - 1 \leq {}^{l(C)}/_{w(C)}$, thus $\lfloor {}^{l(C)}/_{w(C)} \rfloor = k - 1$.

Acknowledgement

The authors would like to express their sincere gratitude to Yusuke Kobayashi for his support in the early stage of this research.

This research was partially supported by the National Research, Development and Innovation Fund of Hungary (grant No. 124171, 128062, 134604, 135606, and FK 132524), and also supported by the János Bolyai Research Scholarship of the Hungarian Academy of Science.. The research reported in this paper was supported by the BME Artificial Intelligence TKP2020 IE grant of NKFIH Hungary (BME IE-MI-SC TKP2020). Research supported in part by National Research, Development and Innovation Office - Grant No. NKFI-115288. Supported by the ÚNKP-22-4-II-BME-248 New National Excellence Program of the Ministry for Culture and Innovation from the source of the National Research, Development and Innovation Fund. Application Domain Specific Highly Reliable IT Solutions" project has been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the Thematic Excellence Programme TKP2020-NKA-06 (National Challenges Subprogramme) funding scheme.

References

[1] Shaping Europe's digital future, Actors in the broadband value chain. European Commission, `https://ec.europa.eu/digital-single-market/en/actors-broadband-value-chain`. Accessed: 2019.

[2] O. Crochat, J.-Y. Le Boudec, and O. Gerstel. Protection interoperability for WDM optical networks. *IEEE/ACM Trans. Netw.*, 8(3):384–395, 2000.

[3] O. Gerstel, M. Jinno, A. Lord, and S. B. Yoo. Elastic optical networking: A new dawn for the optical layer? *IEEE Commun. Mag.*, 50(2):s12–s20, 2012.

[4] M. F. Habib, M. Tornatore, M. De Leenheer, F. Dikbiyik, and B. Mukherjee. Design of disaster-resilient optical datacenter networks. *J. Lightw. Technol.*, 30(16):2563–2573, 2012.

[5] J.-Q. Hu. Diverse routing in optical mesh networks. *IEEE Trans. Communications*, 51:489–494, 2003.

[6] Y. Kobayashi and K. Otsuki. Max-flow min-cut theorem and faster algorithms in a circular disk failure model. In *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, pages 1635–1643, April 2014.

[7] C. MacDiarmid, B. Reed, and L. Schrijver. Non-interfering dipaths in planar digraphs. Jan. 1991.

[8] C. Mcdiarmid, B. Reed, A. Schrijver, and B. Shepherd. Induced circuits in planar graphs. *Journal of Combinatorial Theory, Series B*, 60(2):169 – 176, 1994.

[9] Y. Nemoto and K. Hamaguchi. Resilient ICT research based on lessons learned from the Great East Japan Earthquake. *IEEE Commun. Mag.*, 52(3):38–43, 2014.

[10] S. Neumayer, A. Efrat, and E. Modiano. Geographic max-flow and min-cut under a circular disk failure model. *Computer Networks*, 77:117–127, 2015.

[11] S. Neumayer, G. Zussman, R. Cohen, and E. Modiano. Assessing the vulnerability of the fiber infrastructure to disasters. *IEEE/ACM Trans. Netw.*, 19(6):1610–1623, 2011.

[12] K. Otsuki, Y. Kobayashi, and K. Murota. Improved max-flow min-cut algorithms in a circular disk failure model with application to a road network. *European Journal of Operational Research*, 248(2):396–403, 2016.

[13] L. Shen, X. Yang, and B. Ramamurthy. Shared risk link group (SRLG)-diverse path provisioning under hybrid service level agreements in wavelength-routed optical mesh networks. *IEEE/ACM Transactions on networking*, 13(4):918–931, 2005.

[14] J. W. Suurballe. Disjoint paths in a network. *Networks*, 4:125–145, 1974.

[15] J. Tapolcai, L. Rónyai, B. Vass, and L. Gyimóthi. List of shared risk link groups representing regional failures with limited size. In *IEEE INFOCOM*, Atlanta, USA, May 2017.

[16] B. Vass, E. Bérczi-Kovács, A. Barabás, Z. L. Hajdú, and J. Tapolcai. Polynomial-time algorithm for the regional SRLG-disjoint paths problem. In *Proc. IEEE INFOCOM*, London, United Kingdom, May 2022.

[17] B. Vass, J. Tapolcai, and E. Bérczi-Kovács. Enumerating maximal shared risk link groups of circular disk failures hitting k nodes. *IEEE Transactions on Networking*, 2021.

[18] B. Vass, J. Tapolcai, Z. Heszberger, J. Bíró, D. Hay, F. A. Kuipers, J. Oostenbrink, A. Valentini, and L. Rónyai. Probabilistic shared risk link groups modelling correlated resource failures caused by disasters. *IEEE Journal on Selected Areas in Communications (JSAC) - issue on Latest Advances in Optical Networks for 5G Communications and Beyond*, 2021.

Quest for graphs of Frank number 3

János Barát[1]

Alfréd Rényi Institute of Mathematics
University of Pannonia, Department of Mathematics
8200 Veszprém, Egyetem utca 10., Hungary
barat@renyi.hu

Zoltán L. Blázsik[2]

Alfréd Rényi Institute of Mathematics
MTA–ELTE Geometric and Algebraic Combinatorics Research Group
University of Szeged
1053, Budapest, Reáltanoda u. 13-15., Hungary
blazsik@renyi.hu

Abstract: In an orientation O of the graph G, the edge e is deletable if and only if $O - e$ is strongly connected. For a 3-edge-connected graph G, Hörsch and Szigeti defined the Frank number as the minimum k for which G admits k orientations such that every edge e of G is deletable in at least one of the k orientations. They conjectured the Frank number is at most 3 for every 3-edge-connected graph G. They proved the Petersen graph has Frank number 3, but this was the only example with this property. We show an infinite class of graphs having Frank number 3. Hörsch and Szigeti showed every 3-edge-colorable 3-edge-connected graph has Frank number at most 3. It is tempting to consider non-3-edge-colorable graphs as candidates for having Frank number greater than 2. Snarks are sometimes a good source of finding critical examples or counterexamples. One might suspect various snarks should have Frank number 3. However, we prove several candidate infinite classes of snarks have Frank number 2. As well as the generalized Petersen Graphs $GP(2s+1, s)$. We formulate numerous conjectures inspired by our experience. The full version of this paper can be found in [3].

Keywords: Frank number, strong orientation, 3-edge-connected graphs, snarks

1 Introduction

The graphs in this extended abstract are finite and without loops or multiple edges. We recommend the book by Bondy and Murty [1] for the concepts and notations used here.

A graph G is defined by its vertex set V and edge set E. An *orientation* of G is a directed graph $D = (V, A)$ such that each edge $uv \in E$ is replaced by exactly one of the arcs (u, v) or (v, u).

A *circuit* is a directed cycle. A graph is *cubic* if every vertex has degree 3. A *chord* of a cycle or circuit $v_1, \ldots, v_k$ is an edge connecting two non-consecutive vertices. A graph is 3-*edge-connected* if and only if the removal of any two edges leaves a connected graph.

A directed graph is *strongly connected* if and only if selecting an ordered pair (x, y) of vertices, there is a directed (x, y)-path. An orientation of G is k-*arc-connected* if and only if the removal of any $k - 1$ arcs leaves a strongly connected directed graph.

Theorem 1 (Robbins) *A graph has a strongly connected orientation if and only if it is* 2*-edge-connected.*

The following theorem is a fundamental result in the theory of directed graphs [4].

[1]Research supported by ERC Advanced Grant "GeoScape" and the National Research, Development and Innovation Office, grant K-131529.

[2]The author was supported by the ÚNKP-22-4-SZTE-480 New National Excellence Program of the Ministry for Culture and Innovation from the source of the National Research, Development and Innovation Fund. The research was supported by the Hungarian National Research, Development and Innovation Office, OTKA grant no. SNN 132625.

Theorem 2 (Nash-Williams) *A graph has a k-arc-connected orientation if and only if it is $2k$-edge-connected.*

This opens the question for orientations of 3-edge-connected graphs. This was the motivation for Hörsch and Szigeti [2] for the following concepts. In an orientation O of G the edge e is *deletable* if and only if $O - e$ is strongly connected. For a 3-edge-connected graph G, Hörsch and Szigeti defined the *Frank number* $F(G)$ as the minimum k for which G admits k orientations such that every edge e of G is deletable in at least one of the k orientations. Hörsch and Szigeti [2] showed that any 3-edge-connected graph G satisfies $F(G) \leq 7$ improving on an earlier result.

They also showed any 3-edge-colorable G has Frank number at most 3, and the Petersen graph has Frank number 3. These results made us think probably some other non-3-edge-colorable graphs might have Frank number larger than 2. Snarks are 4-edge-chromatic cubic graphs and usually their girth is at least 5. The Petersen graph is the smallest snark. The next smallest are the Blanuša snarks. We will show that Blanuša snarks have Frank number 2. We also studied an infinite snark family. We will show that each Flower snark has Frank number 2.

Some crucial properties of the Petersen graph can be generalized to the so called Generalized Petersen graphs $GP(2s+1, s)$. One might hope to find a graph among those, which has Frank number 3. However, we will prove that $F(GP(2s+1, s)) = 2$ for $s \geq 3$.

These results lead to the question whether there are any graphs with Frank number greater than 2 besides the Petersen graph. As our main result, we construct infinitely many graphs with Frank number 3. We show an operation, which preserves the Frank number and the edge-connectivity of 3-edge-connected graphs, and produces a cubic graph from a cubic graph. A graph H is a *truncation* of a cubic graph G if a vertex v of G is replaced by a triangle v_1, v_2, v_3 such that each neighbour of v is adjacent to one of v_1, v_2, v_3 so that H remains cubic. Truncation was probably first used in connection with Hamiltonian cycles of polyhedra. In the next section, we introduce the *local cubic modification*, which generalizes truncation to vertices of larger degree.

Theorem 3 *There are infinitely many cubic graphs G such that $F(G) = 3$. They can be constructed from the Petersen graph by successive truncations.*

For instance, the first truncation of the Petersen graph is the Tietze graph. We will prove exhaustively that indeed the Petersen graph is the only cubic 3-edge-connected graph on at most 10 vertices having Frank number 3. Inspired by the proofs of Hörsch and Szigeti, we can show the following.

Theorem 4 *Let G denote a 3-edge-connected graph such that $F(G) \geq 3$. Then there exists a cubic triangle-free graph H^* such that $F(H^*) \geq F(G) \geq 3$.*

2 Preliminaries

If O is an orientation of G, then let $-O$ be the orientation, which we get by reversing every arc in O.

Fact 5 *The set of deletable edges is the same for O and $-O$.*

We routinely have to check if an edge is deletable. The following observation shows one way to do that.

Proposition 6 *Let G be an arbitrary 2-edge-connected graph, and $e = uv \in E(G)$. Suppose that O is a strongly connected orientation of G such that the arc corresponding to e goes from u to v. The orientation $O - e$, which we get by deleting the arc (u, v) from O, is strongly connected if and only if there exists a directed path in $O - e$ from u to v.*

Proof: If there is no (u, v)-path in $O - e$, then $O - e$ is not strongly connected by definition.

If there is a (u, v)-path P in $O - e$, then in any (x, y)-path of O, which uses the arc (u, v), we replace (u, v) by P. Since O was strongly connected, we now find an (x, y)-walk in $O - e$ for any pair x and y. Therefore $O - e$ is strongly connected. □

Let us remark that Proposition 6 is true even if the edge e is contained in an edge cut C of size 2. In this case, $O - e$ cannot admit a strongly connected orientation and we can deduce this by showing that there are no directed paths from u to v. Suppose to the contrary $O - e$ contains a directed path from u to v. Consequently, C must be a directed cut contradicting that O is a strongly connected orientation.

Fact 7 *Suppose G is a graph and its strongly connected orientations $O_1, O_2, \ldots, O_k$ show $F(G) = k$. By the strong connectivity, there is no sink or source of degree 3 in O_i, for any $i \in \{1, 2, \ldots, k\}$.*

In a directed graph, a vertex x of total degree 3 is red, if there are precisely two arcs leaving x, similarly green, if there are precisely two arcs entering x. The following observation gives a necessary but not sufficient condition on the deletability of an arc in a cubic graph.

Fact 8 *If G is a cubic graph and O is a strongly connected orientation of G, then an arc $e = (u, v)$ can be deletable only if u is red and v is green.*

By Proposition 6, the deletability of the arc (u, v) is equivalent to the existence of a directed path from u to v in $O - e$. Therefore u must have outdegree exactly 2, and v must have indegree exactly 2. However, the example in Figure 1 shows that these degree conditions are insufficient. If there exists an edge cut containing e such that every arc except e are going in the same direction, then after deleting e, this edge cut becomes a directed cut, hence no directed (u, v)-path exist anymore regardless of the in- and outdegree of u and v.

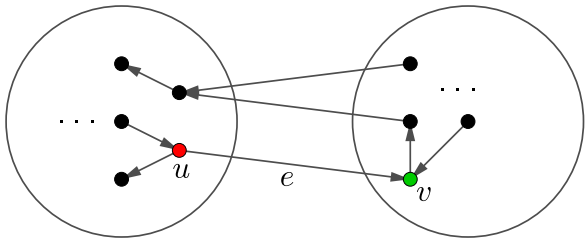


Figure 1: The arc $e = (u, v)$ is not deletable despite the fact that u is red, and v is green

We use the following observation repeatedly. If O is a strongly connected orientation of a 2-edge-connected graph and C is a circuit of O, then every chord of C is deletable regardless of its orientation. Thus if O contains a Hamiltonian circuit C, then every arc of $O - C$ is deletable.

2.1 Three elementary classes

We briefly state our results on three elementary classes of graphs without the proofs. For the detailed proofs, see [3].

For a positive integer $n \geq 3$, the wheel W_n consists of a hub vertex v_0 and n other vertices forming a cycle such that v_0 is adjacent to all other vertices forming the spoke edges. Notice that W_3 is the complete graph on 4 vertices.

Lemma 9 *For every positive integer $n \geq 3$, the wheel W_n has Frank number 2.*

For an even integer $n \geq 4$, let the Möbius ladder M_n be defined as follows. Let $v_1, \ldots, v_n$ be a cycle and we connect each opposite pair, these are edges of form $v_i v_{i+n/2}$.

Lemma 10 *For every positive even integer $n \geq 4$, the graph M_n has Frank number 2.*

Lemma 11 *For every k, the prism $P_k = C_k \times K_2$ has Frank number 2, where $k \geq 3$.*

3 Main result

Hörsch and Szigeti [2] introduced the notion of *cubic extensions* of a graph with minimum degree at least 3 in Subsection 2.3. It is a global modification, which replaces every vertex v of degree at least 4 with a cycle of size $deg(v)$, leave the vertices of degree 3 intact, and substitute every edge with an edge between the corresponding objects in such a way that this not necessarily unique graph is cubic.

In contrast to that, we use the following local operation on a graph G of minimum degree at least 3. For $d \geq 3$, let v be a vertex of degree d, and let the neighbours of v be $x_1, \ldots, x_d$. We remove v and replace each edge vx_i by an edge v_jx_i and add a cycle C_v on $v_1, \ldots v_d$ (see Figure 2) so that each of the new vertices has exactly one neighbour from $x_1, x_2, \ldots, x_d$. The resulting graph G_v is a *local cubic modification* of G at v. Let us remark that G_v is not necessarily unique, it depends on the chosen perfect matching between $\{x_1, x_2, \ldots, x_d\}$ and $\{v_1, v_2, \ldots, v_d\}$. Note that for $d = 3$ the truncation is a special local cubic modification.

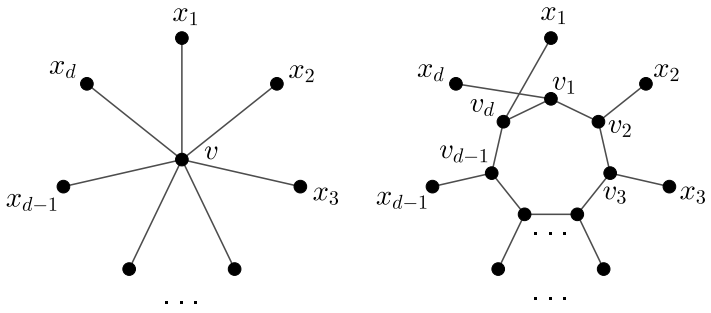


Figure 2: A local cubic modification at v

Let us emphasize that at this point it may happen that after performing a local cubic modification the edge-connectivity decreases (see Figure 3).

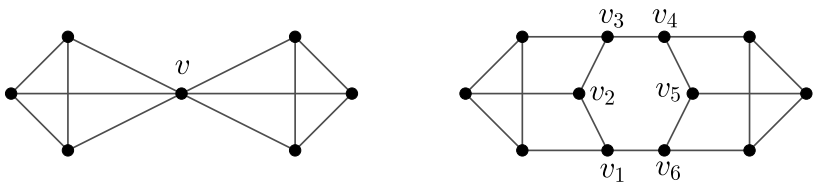


Figure 3: The edge-connectivity may decrease by performing a local cubic modification at v

It is not trivial, but for every 3-edge-connected graph we can show that there exists a local cubic modification, which remains 3-edge-connected. Therefore in the sequel, we assume the local cubic modification G_v of the 3-edge-connected graph G at vertex v is always 3-edge-connected. However, it is true that every cubic extension of a graph can be realized as a series of local cubic modifications, and in the other direction if we perform a series of local cubic modifications of a graph at all vertices of degree at least 4, then we get a cubic extension. Consequently, the previous observation means that one can find a 3-edge-connected cubic extension of a 3-edge-connected graph even if there are cut vertices.

The following general observation plays a key role in the next proofs when applied to local cubic modification.

Fact 12 *Let G_v be the local cubic modification of G at v, and an orientation O_v is given such that there exists a directed (y, z)-path P_v^{yz} in O_v for $\{y, z\} \not\subseteq \{v_1, v_2, \ldots, v_d\}$. Now a directed (y, z)-path also exists for the inherited orientation O of G for the corresponding (y, z) pair.*

Now we are ready to show that a local cubic modification cannot decrease the Frank number. Moreover, if the vertex v has degree 3, then it cannot increase either. Hence in that case, the Frank number remains the same.

Lemma 13 *Let G be a 3-edge-connected graph. If G_v is a local cubic modification of G at v, then $F(G_v) \geq F(G)$.*

PROOF: Suppose to the contrary that $F(G_v) = k < F(G)$ witnessed by the strongly connected orientations $O_1^v, \ldots, O_k^v$. Let $O_1, \ldots, O_k$ be the orientations of G, which coincide with $O_1^v, \ldots, O_k^v$ on identical edges. Also let the direction of $v_j x_i$ be copied to vx_i in each orientation. Since each O_j^v was strongly connected, for any pair of vertices y, z there exists a directed path between them in both directions. By Fact 12, we can deduce that O_j also has the same property hence it is strongly connected.

We claim each edge $e = yz$ of G is deletable in at least one orientation. Let O_j^v be the orientation of G_v, where e with the appropriate orientation (say (y, z)) was deletable. We know that O_j^v is strongly connected and contains a directed (y, z)-path P_{yz}^v in $O_j^v - \{e\}$. Consequently, similarly to the proof of Fact 12, $O_j - \{e\}$ contains a directed (y, z)-path since in P_{yz}^v we can contract the part between the first and last appearance of some v_i for an appropriate i. Therefore e is deletable in O_j by Proposition 6. □

Corollary 14 *Let G be a 3-edge-connected graph. There exists a cubic extension H of G, which is 3-edge-connected and $F(H) \geq F(G)$.*

By Lemma 13, we can create an infinite family $\mathcal{G}$ of cubic graphs with $F(G) \geq 3$ for any $G \in \mathcal{G}$ starting from the Petersen graph in the following way. Hörsch and Szigeti [2] showed the Petersen graph has Frank number 3. Pick an arbitrary vertex v of the Petersen graph, and consider the local cubic modification G_v of G at v. Since the Petersen graph is cubic and 3-edge-connected and G_v is 3-edge-connected as well, hence by Lemma 13, we get $F(G_v) \geq F(G)$. After iterating this local cubic modification procedure with an arbitrary vertex of the always cubic current graph, the Frank number never decreases. Thus we created an infinite family of 3-edge-connected graphs with Frank number at least 3.

In Theorem 3, we claimed the existence of an infinite family of cubic graphs with Frank number equal to 3. So far we have seen that the Frank number cannot decrease performing a local cubic modification at an arbitrary vertex v. In the next Lemma, we show that the Frank number cannot increase if $deg(v) = 3$.

Lemma 15 *Let G be a 3-edge-connected graph and v a vertex of degree 3. If G_v is a local cubic modification of G at v, then $F(G_v) \leq F(G)$.*

PROOF: Suppose the orientations $\mathcal{O} = \{O_1, O_2, \ldots, O_k\}$ are the witnesses of $F(G) = k$. We create k orientations $\mathcal{O}^v = \{O_1^v, O_2^v, \ldots, O_k^v\}$ of G_v to prove $F(G_v) \leq k$. Let us focus on the truncated part of G_v, we just copy the orientations from the corresponding O_i outside of the modified part.

Since every O_i is a strong orientation, the 3-edge-cut formed by edges $\{av, bv, cv\}$ cannot be a directed cut. By Fact 5, we might assume that in every orientation O_i, exactly two edges leave v. For convenience, instead of referring to a, b, c as the concrete neighbours of v, let us permute their roles. We may assume that a denotes the tail of the unique arc entering v. In Figure 4, we introduce the four orientations we use later in this proof. Note that the first two orientations become the same if we interchange the roles of b and c, and so do the last two orientations. Hence there are essentially two types of extensions which we use on the truncated part of G_v.

Firstly, observe that no matter which extensions we use from Figure 4, the orientation O_i^v we get is also strongly connected. Indeed, we can enter the triangle v_a, v_b, v_c only from a and we can leave in both directions through b or c, hence every directed path of O_i can be extended even if it goes through v in G. Moreover, there exists a directed path between any pair of new vertices in O_i^v.

An arc of O_i not incident to v is deletable if and only if the same arc is deletable in O_i^v. By Proposition 6, it is enough to show a directed path between its endpoints in the modified graph as well. As we discussed in the previous paragraph, this can be done and it does not depend on the choice of the orientation of

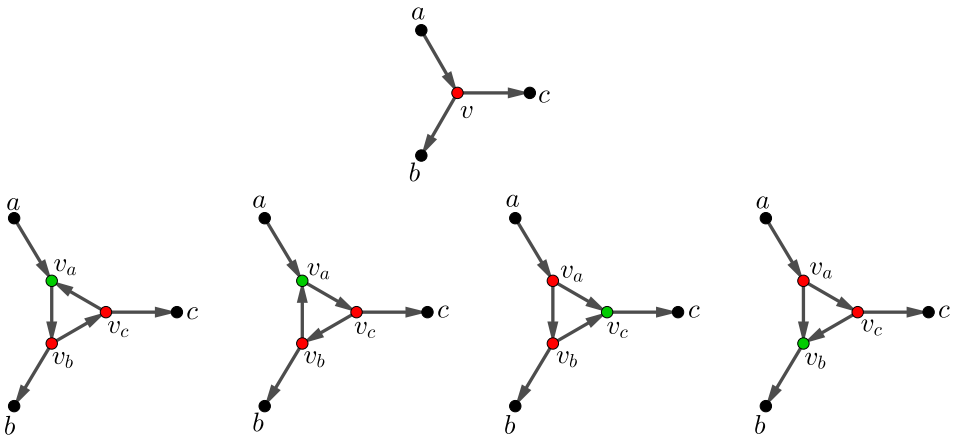


Figure 4: The four orientations we use on the new arcs (essentially two different types)

the triangle at the truncated vertex v as long as we use the four orientations above. Therefore for every edge not incident to v, there exists an orientation O_i^v of G_v so that the corresponding arc is deletable in O_i^v.

Choose a smallest subset $\mathcal{S} = \{O_{j_1}, O_{j_2}, \ldots, O_{j_\ell}\}$ of $\mathcal{O}$ such that all of the edges incident to v is deletable in at least one of the orientations in $\mathcal{S}$. Here $1 < \ell \leq 3$ holds.

If $|\mathcal{S}| = 2$, then in at least one of these orientations both arcs leaving v are deletable and in the other orientation the third edge incident to v is not just outgoing but also deletable. In Figure 5, we show how the orientations $\{O_{j_1}^v, O_{j_2}^v\}$ look like at the truncated vertex v (remember that the role of b and c are interchangeable). Notice that the blue color and also the X marks (for the black and white versions) indicate which arcs are deletable.

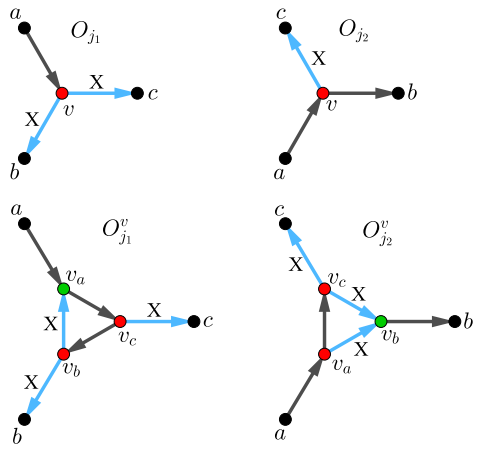


Figure 5: The orientations $\{O_{j_1}^v, O_{j_2}^v\}$, if $|\mathcal{S}| = 2$

Indeed, the arcs of type (v_x, x) are deletable in $O_{j_i}^v$ if and only if (v, x) was deletable in O_{j_i}. The arcs inside the triangle of type (v_x, v_y) are deletable either trivially or because of the fact that O_{j_i} is strongly connected.

If $|\mathcal{S}| = 3$, then for each of the edges incident to v there is a unique orientation of $\mathcal{S}$ so that the corresponding arc is deletable. Using any of the last two orientations in Figure 4 results in three orientations for which every arc of the triangle is also deletable in at least one of them. Indeed, the arc opposite to

the deletable one which leaves v is always deletable by Proposition 6 since there is a directed path within the triangle.

Naturally, we can use any of the orientations described in Figure 4 in any of those orientations of $\mathcal{O}$ which haven't been touched yet. Hence we proved that $F(G_v) \leq F(G)$. □

Corollary 16 *Lemma 13 and Lemma 15 together implies that if a 3-edge-connected graph G contains at least one vertex of degree 3, then by successively performing a local cubic modification at vertices of degree 3 we get a family of graphs with the same Frank number as G. Notice that in each step, the newly introduced vertices have degree 3.*

Thus if we start with the Petersen graph, we can build a family of graphs with Frank number exactly 3 concluding the proof of Theorem 3. However, if a graph H contains a triangle T, then we can contract the vertices of T into a new vertex v_T (or in other words identify these vertices) such that v_T is adjacent to the other neighbours of the three vertices of T, thus the resulting graph H/T is simple (since H was cubic) and cubic.

What can we say about the relation between the Frank number of H and H/T?

Since H is a local cubic modification of H/T at v_T, we get $F(H) \geq F(H/T)$ by Lemma 13. On the other hand, Lemma 15 yields that $F(H/T) \geq F(H)$ since v_T is a vertex of degree 3 in H/T. Hence $F(H) = F(H/T)$. Consequently, we can contract triangles starting from H until the resulting graph H^* is either triangle-free or $H^* \simeq K_4$ while the Frank number remains the same. We know that $F(K_4) = 2$, and $F(H^*) \geq 2$ if H^* is a 3-edge-connected cubic triangle-free graph.

Proof:(Proof of Theorem 4) By Corollary 14, we can consider the cubic extension H of G for which $F(H) \geq F(G)$. Then after successively contracting triangles the resulting graph H^* is either triangle-free or it is K_4 while $F(H^*) = F(H)$. Since $F(G) \geq 3$ thus $H^* = K_4$ is a contradiction, hence we get a 3-edge-connected cubic triangle-free graph H^* such that $F(H^*) \geq F(G) \geq 3$. □

This result may help the computer aided search for other 3-edge-connected graphs with higher Frank number.

4 Snarks

Snarks are bridgeless cubic graphs with chromatic index 4. The Petersen graph is the smallest such graph. Hörsch and Szigeti [2] proved each 3-edge-connected, 3-edge-colorable graph has Frank number at most 3, and the Petersen graph has Frank number 3. Therefore, we expected to find other examples with Frank number 3 among snarks.

In this section, we investigate the second smallest snarks that are the Blanuša snarks and an infinite family of snarks, the so-called flower snarks. For every odd $n \geq 3$ let J_n denote the flower snark on $4n$ vertices. One can construct this graph starting with n copies of stars on 4 vertices with centers $v_1, v_2, \ldots, v_n$ and outer vertices denoted by $\{a_i, b_i, c_i\}$ for $1 \leq i \leq n$. Then add an n-cycle on the vertices $(a_1, a_2, \ldots, a_n)$, and a $2n$-cycle on $(b_1, b_2, \ldots, b_n, c_1, c_2, \ldots, c_n)$.

It turns out that the Frank number of each of these snarks is 2. The proofs for the two types of snarks are very similar, and we handle them together.

Theorem 17 *Both Blanuša snarks, and every flower snark has Frank number 2.*

Proof: Since these snarks are not 4-edge-connected, therefore they do not admit a 2-arc-connected orientation by Theorem 2. Hence their Frank number must be greater than 1. On the other hand, we show two strongly connected orientations $\{O_1, O_2\}$ of these snarks in Figures 6, 7, 8 that verify that their Frank number is at most 2, which concludes the proof.

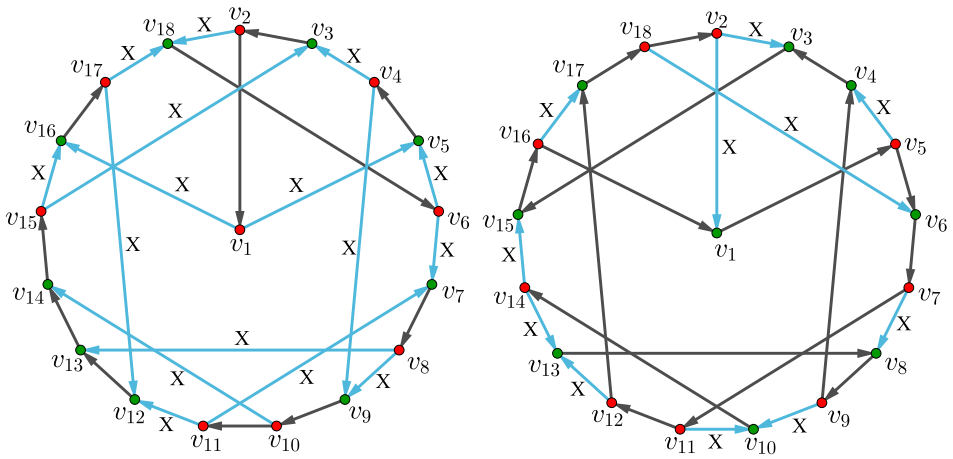


Figure 6: The first Blanuša snark has Frank number 2

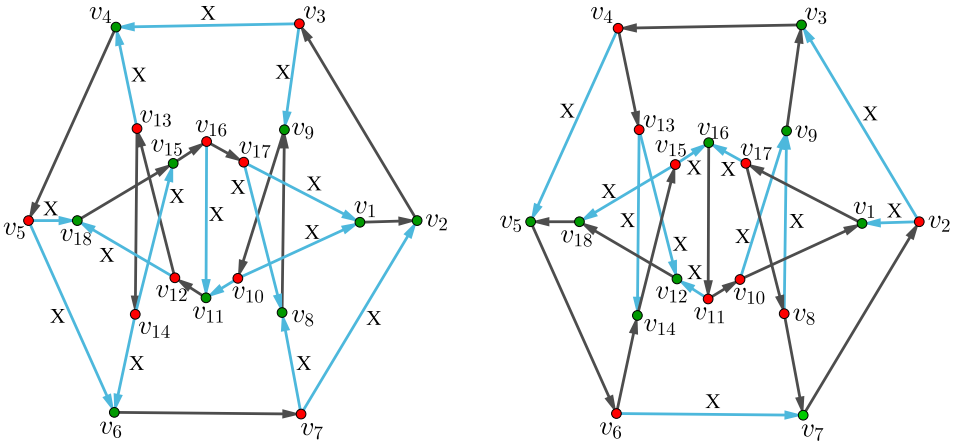


Figure 7: The second Blanuša snark has Frank number 2

The first thing is to check that these orientations are indeed strongly connected. To see this, observe in each orientation, each vertex is covered by a circuit. For any two vertices there is a chain of intersecting circuits covering these vertices, hence there is a directed path between them in both directions.

To prove that an arc (u, v) is deletable, it is enough to find a directed path from u to v after the deletion of (u, v) by Proposition 6.

In Figures 6, 7, 8 the blue arcs (also marked by X) indicates the deletable arcs of the corresponding orientations. Some hints are included in Figure 8 which can be generalized for an arbitrary flower snark J_n. However, for the two Blanuša snarks, there is no general rule (other than using the still intact circuits) for deciding whether an arc is deletable or not, one should manually check them. But finding the appropriate directed path after the deletion is usually straightforward due to the small degrees of the vertices. □

5 Generalized Petersen graphs

We investigated the most natural generalized Petersen graphs in the hope of finding another example of a 3-edge-connected graph with Frank number at least 3. As it turned out, the generalized Petersen graph

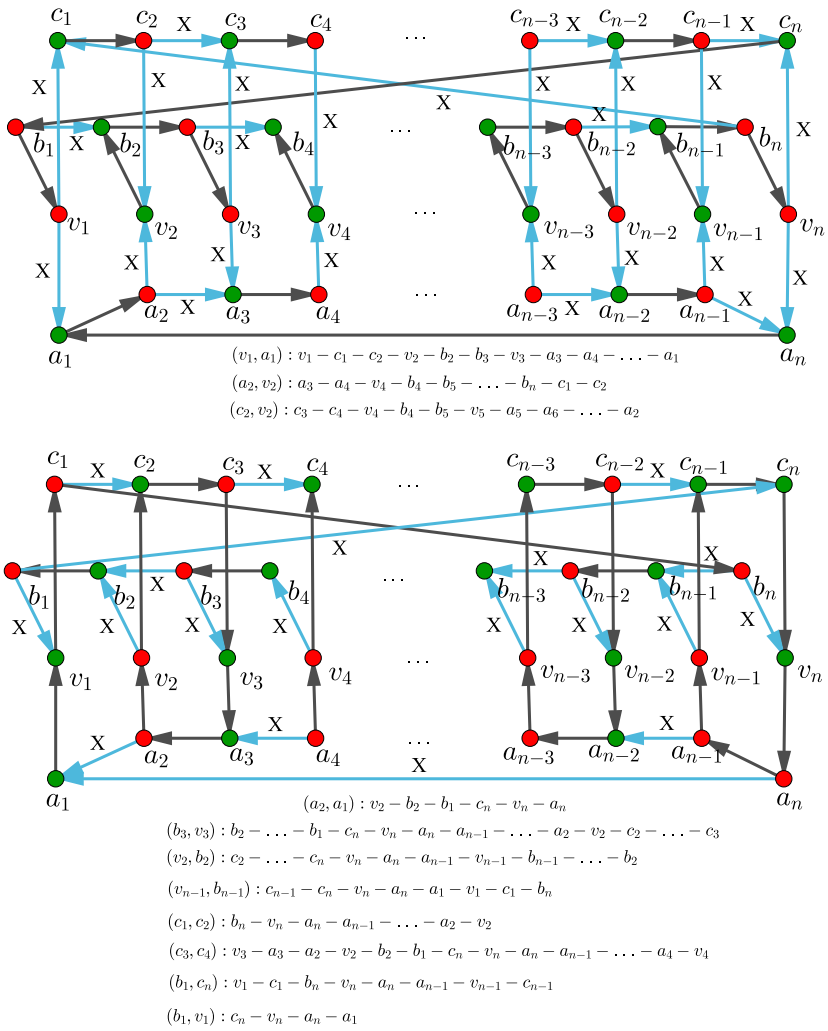


Figure 8: $F(J_n) = 2$, for any odd n

$G(2s+1, s)$ admits two appropriate orientations, consequently its Frank number is 2.

Theorem 18 *If $GP(2s+1, s)$ denotes the generalized Petersen graph for $s \geq 3$, then $F(GP(2s+1, s)) = 2$.*

Instead of the detailed proof, we just illustrate the two orientations in Figure 9, 10 for small values.

Discussion

We pose the following conjectures, each of which is relaxing the strong conjecture that every 3-edge-connected graph has Frank number at most 3.

Conjecture 19 *For every cubic 3-edge-connected graph G, there exists a strongly connected orientation D of G such that for every vertex v, there exists an arc a_v incident to v such that $D - a_v$ is strongly connected.*

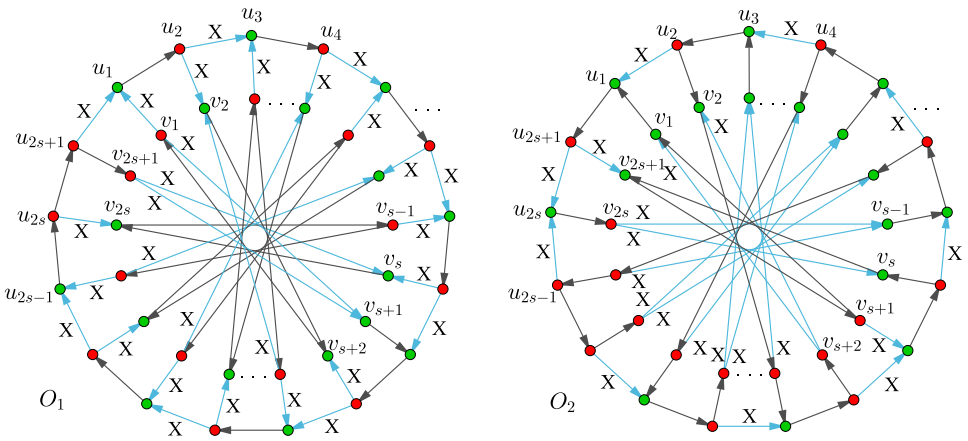


Figure 9: $F(G(2s+1,s)) = 2$ for $s \geq 3$, s even (illustrated for $s = 8$)

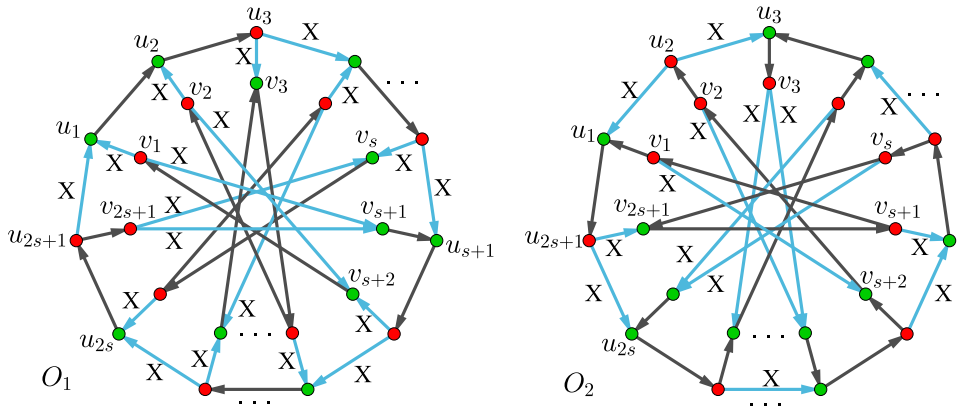


Figure 10: $F(G(2s+1,s)) = 2$ for $s \geq 3$, s odd (illustrated for $s = 5$)

Conjecture 20 *For every cubic 3-edge-connected graph G, there exists a strongly connected orientation D of G such that for at least half of the arcs $D - a$ is strongly connected.*

Conjecture 21 *If a 3-edge-connected cubic graph G admits a Hamiltonian cycle, then G has Frank number 2.*

References

[1] J.A. Bondy, U.S.R. Murty. Graph Theory, *Springer-Verlag*, London, XII+663 pages, (2008).

[2] F. Hörsch, Z. Szigeti. Connectivity of orientations of 3-edge-connected graphs. *European J. Combin.* **94** (2021).

[3] J. Barát, Z. L. Blázsik Quest for graphs of Frank number 3. *arXiv version* https://arxiv.org/abs/2209.08804 (2022).

[4] C.St.J.A. Nash–Williams. On orientations, connectivity, and odd vertex pairings in finite graphs. *Canad. J. Math.*, **12**:555–567, (1960).

Arc-partitioning and vertex-ordering problems

Nóra Anna Borsik

Department of Operations Research, ELTE Eötvös Loránd University, Pázmány Péter sétány 1/C, 1117 Budapest, Hungary.
borsiknora@student.elte.hu

Péter Madarasi

Department of Operations Research, ELTE Eötvös Loránd University, and the ELKH-ELTE Egerváry Research Group on Combinatorial Optimization, Eötvös Loránd Research Network (ELKH), Pázmány Péter sétány 1/C, 1117 Budapest, Hungary.
madarasip@staff.elte.hu

Abstract: The fundamental question of this paper is deciding whether all directed cycles of a digraph can be covered with certain types of subgraphs, that is, whether a feedback arc set with certain properties exists. For example, deciding whether an in-branching feedback arc set exists will be shown to be polynomial-time solvable. However, partitioning into a matching and an acyclic subgraph turns out to be NP-hard.

Generalizing the case of in-branchings, we introduce the (f, g)-*FAS problem*, in which our goal is to decide whether an order of the vertices exists such that the left (weighted) out-degree of each vertex v is at least $f(v)$ and at most $g(v)$, where f and g are lower and upper bound functions on the vertices. We show that this problem is NP-complete, but it is polynomial-time solvable if either only lower or upper bounds are given on the vertices, which — as a special case — solves the problem of partitioning into an in-branching and an acyclic subgraph. The algorithm is described for a much more general version of the problem in which arbitrary non-decreasing set-functions play the role of the left out-degrees. Some natural modifications of the upper-bounded case also turn out to be NP-hard, for example, if for a single vertex both lower and upper bounds are given. The (f, g)-FAS problem with bounds $f(v) = 1$ and $g(v) = \delta(v) - 1$ for each vertex v except the first and the last ones, is equivalent to the so-called s-t numbering problem, which is known to be polynomial-time solvable. However, the (f, g)-FAS problem becomes NP-complete basically with any stricter bounds, that is, for any parameters $a \geq 1$ and $b \geq 2$ with bounds $f(v) = a$ and $g(v) = \delta(v) - b$ for each vertex v except the first and the last ones.

Keywords: Arc partitioning, Vertex ordering, Feedback arc set, Graph decomposition, Rank aggregation, NP-completeness

1 Introduction

The fundamental question of this paper is deciding whether all directed cycles of a digraph can be covered with certain types of subgraphs, that is, whether a feedback arc set with certain properties exists. We are going to show that partitioning into an in-branching and an acyclic subgraph is polynomial-time solvable, where an in-branching is a directed forest in which every weakly connected component C is directed towards a specified root vertex $r \in C$. However, partitioning into a matching and an acyclic subgraph is NP-complete.

[1]This research has been implemented with the support provided by the Ministry of Innovation and Technology of Hungary from the National Research, Development and Innovation Fund, financed under the ELTE TKP 2021-NKTA-62 funding scheme. The research was supported by the Ministry of Innovation and Technology NRDI Office within the framework of the Artificial Intelligence National Laboratory Program.

First, we introduce the $(f, g; \sum w)$-FAS problem, which generalizes the problem of partitioning a digraph into an in-branching and an acyclic subgraph. In addition, the (f, g)-FAS problem can be used to solve some rank aggregation problems, as we will see at the end of this section.

Problem 1 *Let us given a loop-free digraph $D = (V, A)$ with a weight function $w : A \to \mathbb{R}_+$ on the arcs, and lower and upper bound functions $f : V \to \mathbb{R}_+$ and $g : V \to \mathbb{R}_+$ on the vertices. Our goal is to decide whether there exists an order of the vertices such that*

$$f(v) \leq \overleftarrow{\delta}^w(v) \leq g(v)$$

holds for every vertex $v \in D$, where $\overleftarrow{\delta}^w(v)$ denotes the left weighted out-degree of v according to the order. We call this problem the $(f, g; \sum w)$-FAS problem*, where FAS stands for feedback arc set. In the unweighted case, the problem is referred to as the* (f, g)-FAS problem*.*

Let us define a more general ordering problem, which will be shown to be polynomial-time solvable when only upper bounds are present.

Problem 2 *Let us given a ground set V and, for each element $v \in V$, a set-function $h_v : 2^{V-v} \to \mathbb{R}$ which is non-decreasing (that is, $h_v(A) \leq h_v(B)$ holds for all subsets $A \subseteq B \subseteq V - v$). Suppose these functions can be evaluated efficiently. Furthermore, we are given lower and upper bound functions $f : V \to \mathbb{R}$ and $g : V \to \mathbb{R}$. Our goal is to decide whether there exists an order σ of V such that*

$$f(v) \leq h_v(\overleftarrow{\sigma}(v)) \leq g(v)$$

holds for each element $v \in V$, where $\overleftarrow{\sigma}(v)$ denotes the set of the elements preceding v according to the order σ. We call this problem the $(f, g; h)$-ordering problem*, while the feasible orders are called* $(f, g; h)$-orders*.*

Observe that the $(f, g; \sum w)$-FAS problem is a special case of the $(f, g; h)$-ordering problem. To show this, let V be the vertex set of the digraph, and for each vertex v and subset $V' \subseteq V - v$, let

$$h_v(V') = \sum_{e \in \delta(v, V')} w(e),$$

where $\delta(v, V')$ denotes the set of the outgoing arcs from v to V'. By definition, the solutions to the $(f, g; \sum w)$-FAS problem are exactly the $(f, g; h)$-orders.

Arc-partitioning and vertex-ordering problems Arc-partition problems have been studied extensively in the literature [2, 3, 4, 5]. For example, partitioning into a directed cycle and an acyclic subgraph, or a directed cycle factor and an acyclic subgraph are NP-complete problems [2]. It is known that deciding whether a digraph contains an arc-disjoint r-in-arborescence and r-out-arborescence for a given root r is NP-complete, where an r-in-arborescence is an in-branching with only one root vertex r and an r-out-arborescence is a reversed r-in-arborescence. In other words, partitioning into a subgraph containing an r-in-arborescence and a subgraph containing an r-out-arborescence is NP-complete [1]. However, the problem is solvable for two arc-disjoint r-out-arborescences, or for k arc-disjoint r-out-arborescences in general [8]. The well-known feedback arc set problem is about finding the fewest possible arcs whose removal makes the graph acyclic, which is equivalent to finding an order of the vertices minimizing the number of the left-going arcs. The problem is known to be NP-hard [10]. Besides the arc-partition problems, we investigate a new vertex-ordering problem, the $(f, g; \sum w)$-FAS problem. One of its special cases, the so-called s-t numbering problem is about finding an order of the vertices such that each vertex has at least one outgoing arc to the left and at least one to the right. The s-t numbering problem is polynomial-time solvable [7], however, the more general betweenness problem is NP-hard [11].

Rank aggregation problems Consider a competition, where different judges give complete rankings of the candidates and our goal is to find a common ranking which is a "fair" consensus between the judges. In the Kemény rank aggregation problem, the distance of two rankings is defined as the number of those pairs of candidates whose order is reversed in the two rankings. The goal is to find a common ranking minimizing the sum of the distances from the judges' rankings. In another rank aggregation problem, we want to find a ranking which is close to the farthest ranking, so we want to minimize the maximum distance. It is known that both problems are NP-hard [6].

Consider a similar problem, where we define the distance from the viewpoint of the candidates instead of the judges: For a candidate v, let $d(v)$ denote the number of the candidates whom v precedes by the majority of the judges, but not in the common ranking. This is a natural measure how unfair the common ranking seems to the candidate v. Our goal is to find a common ranking minimizing the largest distance $d(v)$ over all candidates.

To reduce the problem to the (f, g)-FAS problem, define a penalty graph in which the vertices correspond to the candidates, and there is an arc from u to v if the majority of the judges rank u before v. If we order the vertices of this graph by an arbitrary ranking, then the left out-degree of v, i.e. the number of arcs from v to the preceding vertices, is equal to the distance $d(v)$ from this ranking. This means that the problem can be rephrased as ordering the vertices of a directed graph such that the maximum left out-degree is minimized. This problem is solvable by finding the smallest positive integer c for which the (f, g)-FAS problem has a feasible solution for $f \equiv -\infty$ and $g \equiv c$.

Our work Section 2.1 solves the $(f, g; h)$-ordering problem in the case when only upper bounds are given. However, the problem becomes NP-complete after some natural modifications, such as when the set-functions are not required to be non-decreasing, or there is a single item with both lower and upper bounds. Section 2.2.1 considers the (f, g)-FAS problem with special bound functions. The problem with bounds $f(v) = 1$, $g(v) = \delta(v) - 1$ on each vertex v except the given first and last vertices s and t, is equivalent to the s-t numbering problem, which is known to be polynomial-time solvable [7]. We are going to show, however, that the problem is NP-complete basically with any stricter bounds, that is, for any parameters $a \geq 1$ and $b \geq 2$ with bounds $f(v) = a$ and $g(v) = \delta(v) - b$ for each vertex v except s and t. By a simple modification of the proof, the problem is also shown NP-complete when $f \equiv g$. In Section 3, we consider arc-partition problems. First, we prove that partitioning into an in-branching and an acyclic subgraph can be solved in polynomial time, moreover, it is also solvable if some vertices are required to be roots in the in-branching. However, the complexity of partitioning into an in-arborescence and an acyclic subgraph remains open. We show that both problems become NP-hard if our goal is to find a minimum-cost in-branching or a minimum-cost in-arborescence whose complement is acyclic. Finally, we give a construction which proves that partitioning into a matching and an acyclic subgraph is NP-complete.

2 Ordering problems

In Section 2.2, the $(f, g; \sum w)$-FAS problem and the $(f, g; h)$-ordering problem will be shown NP-hard in general. The next section, however, proves that even the latter problem can be solved in polynomial time provided that $f \equiv -\infty$ and the function h_v can be evaluated efficiently for all $v \in V$.

2.1 Only upper bounds

By the $(-\infty, g; h)$-ordering problem, we mean the case when only upper bounds are given, that is, $f \equiv -\infty$. This section gives a polynomial-time algorithm for solving this problem. Later, this algorithm and the following theorems will be used to partition a digraph into an in-branching and an acyclic subgraph — or prove that no such partition exists.

Algorithm 1 $(-\infty, g; h)$-ORDERING

```
1: V' := V; n := |V|
2: Let σ_1, ..., σ_n denote the order we are searching for
3: for i = n, ..., 1 do
4:     V* := {v ∈ V' : h_v(V' − v) ≤ g(v)}
5:     if V* ≠ ∅ then
6:         Choose σ_i ∈ V* arbitrarily
7:         V' := V' − σ_i
8:     else
9:         output No solution exists
10:        exit
11:    end if
12: end for
13: output σ_1, ..., σ_n
```

Algorithm 1 fixes the items from right to left. The set of the not-fixed items is denoted by V'. In line 4, the algorithm filters those elements from V' for which $h_v(V') \leq g(v)$. If at least one such element exists, then one of them is selected, placed at the last free position and deleted from V'. If no such item is found, then the algorithm concludes that no solution exists. Next, we show the correctness of Algorithm 1.

Theorem 3 *Algorithm 1 solves the $(-\infty, g; h)$-ordering problem.*

PROOF: Clearly, the fixed items do not violate the upper bounds as $h_v(V' - v) \leq g(v)$ holds whenever an item v is fixed. Hence, if such an item can be selected in every iteration of the for loop, then the algorithm finds a feasible $(-\infty, g; h)$-order. Otherwise, no such item exists and V' is nonempty. Let σ be an arbitrary order of V, and let v be the first item in V' according to this order. Then $\overleftarrow{\sigma}(v) \geq h_v(V' - v) > g(v)$ holds, since $h_v(V' - v) > g(v)$ for all $v \in V'$. Therefore, v violates the upper bound $g(v)$, and σ is not feasible. Hence, no order of V is feasible. □

From the correctness of the algorithm we get the following characterization for the existence of a feasible order:

Theorem 4 *Let us given a ground set V and a non-decreasing set-function h_v for each item $v \in V$. There exists a $(-\infty, g; h)$-order of V if and only if there is no subset $V' \subseteq V$ in which $h_v(V' - v) > g(v)$ holds for each element $v \in V'$.*

As we have already seen, the $(f, g; \sum w)$-FAS problem is a special case of the $(f, g; h)$-ordering problem, hence one gets a characterization for the solvability of the former as a corollary.

Theorem 5 *Let us given a digraph $D = (V, A)$ with a weight function w on its arcs. There exists a solution to the $(-\infty, g; \sum w)$-FAS problem if and only if there is no induced subgraph $D' = (V', A')$ in which $\delta^w(v, V') > g(v)$ holds for each vertex $v \in V'$, where $\delta^w(v, V')$ denotes the weighted out-degree of v restricted to the subgraph D'.*

Note that the $(f, \infty; h)$-ordering problem (in which only lower bounds are given) can be solved with a similar algorithm. The main difference is that it fixes the items from left to right. Furthermore, it is not hard to prove that the $(f, g; h)$-ordering problem remains solvable if there are both lower and upper bounds, but on each vertex either only lower or only upper bound is given.

2.2 Complexity

In this section, the complexities of the $(f, g; \sum w)$-FAS and the $(f, g; h)$-ordering problems are investigated. In the previous section, we showed that both problems can be solved if only upper bounds are present. It is natural to ask whether a similar algorithm exists for more general cases or related problems. We are going to prove that the most natural modifications to the problem make it NP-hard, for example, when we have only upper bounds except for one vertex for which both lower and upper bounds are given.

Theorem 6 *The (f, g)-FAS problem is NP-complete if only upper bounds are given for all vertices except for a single vertex v for which $f(v) = g(v)$.*

PROOF: The (f, g)-FAS problem is clearly in NP. The proof is by reduction from the independent set problem [10]. Let us given a graph $G = (V, E)$ for which we want to solve the independent set problem. Construct the digraph $D = (V_D, A)$ as follows: Let the vertex set of D consist of the vertices and the edges of G, and add a further vertex s. For each edge $e = uv \in E$, let D contain an arc from $e \in V_D$ to $u \in V_D$ and an arc from $e \in V_D$ to $v \in V_D$. Moreover, for each vertex $v \in V$, let D contain an arc from s to $v \in V_D$. Figure 1 illustrates the construction of D.

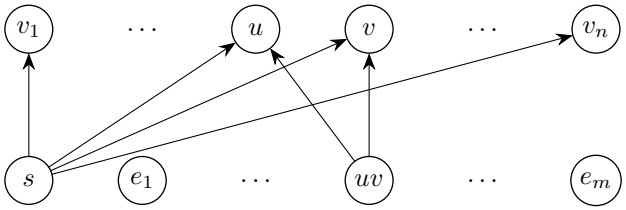


Figure 1: The graph constructed during the reduction from the independent set problem.

In the bottom row of Figure 1, let D contain n parallel arcs from s to e_1 and two parallel arcs from every other vertex e_i to the succeeding vertex e_{i+1}, where $e_1, \ldots, e_m$ are the vertices corresponding to the edges of G. Let $f(s) = g(s) = k$ and $f(v) = -\infty$, $g(v) = 1$ for each vertex $v \in V_D \setminus \{s\}$. We prove that the (f, g)-FAS problem is solvable for D if and only if G has an independent set of size k.

If G has an independent set of size k, then consider the following order of the vertices of D: First, list the vertices of the independent set in arbitrary order, then the vertices in the bottom row of Figure 1 in the order $s, e_1, \ldots, e_m$, and put the remaining vertices in arbitrary order to the end. The resulting order is clearly a feasible solution to the (f, g)-FAS problem defined on D.

Conversely, if there exists a feasible order σ for the (f, g)-FAS problem, then the vertices in the bottom row of Figure 1 must be in the given order, because of the parallel arcs between them. So s precedes all vertices $e \in V_D$ corresponding to the edges of G. The vertex s has bounds $f(s) = g(s) = k$, therefore there must be exactly k vertices before s according to σ. These vertices of D correspond to vertices of G, and they must be independent in G, because the upper bound $g(e) = 1$ for any edge $e \in E \cap V_D$ ensures that only one of its endpoints may precede e. This implies that the first k vertices in σ form an independent set in G, which completes the proof of the theorem. □

In another natural modification, the non-negativity of the arc-weights is not required. The hardness of this problem can be shown similarly to the previous proof.

Theorem 7 *The $(-\infty, g; \sum w)$-FAS problem is NP-complete if negative arc-weights are allowed.*

Corollary 8 *The $(-\infty, g; h)$-ordering problem is NP-complete if the set-functions h_v are not required to be non-decreasing.*

2.2.1 Special bounds

Another interesting question is the complexity of the (f, g)-FAS problem with special bound functions. For example, when the lower and the upper bounds are equal on each vertex, in other words, there is a strict prescription for the left out-degrees of the vertices. The other extreme case is when there is a large difference between the lower and the upper bounds on each vertex. Let us given a first vertex denoted by s and a last vertex denoted by t with bounds $f(s) = g(s) = 0$ and $f(t) = g(t) = \delta(t)$, and on each vertex $v \neq \{s, t\}$, let the bounds be $f(v) = a$ and $g(v) = \delta(v) - b$ for some given non-negative integers a and b. This problem is equivalent to ordering the vertices such that each vertex has at least a outgoing arcs to the preceding vertices and at least b outgoing arcs to the succeeding vertices, except for s and t. If the parameters are $a = b = 1$, then the problem is the so-called s-t numbering problem for directed graphs, which is known to be polynomial-time solvable [7]. In what follows, we prove that the problem is NP-complete with the slightly modified bounds when the parameters are $a = 1$ and $b = 2$, then we extend this result for the case $a \geq 1$, $b \geq 2$.

Theorem 9 *The (f, g)-FAS problem is NP-complete with bounds $f(v) = 1$, $g(v) = \delta(v) - 2$ for each vertex v except the first and the last ones. The problem is NP-complete, even if all out-degrees are at most 3.*

Proof: The proof is by reduction from the NP-complete 3-XSAT-3 problem [12]. Let us given a conjunctive normal form (CNF) formula in which each clause contains exactly 3 literals and each variable is contained in exactly 3 clauses. In the 3-XSAT-3 problem, the goal is to decide whether the formula can be satisfied such that exactly one literal is true in each clause. Let $x_1, \ldots, x_n$ denote the variables and let $c_1, \ldots, c_n$ denote the clauses. We construct an instance of the (f, g)-FAS problem as follows:

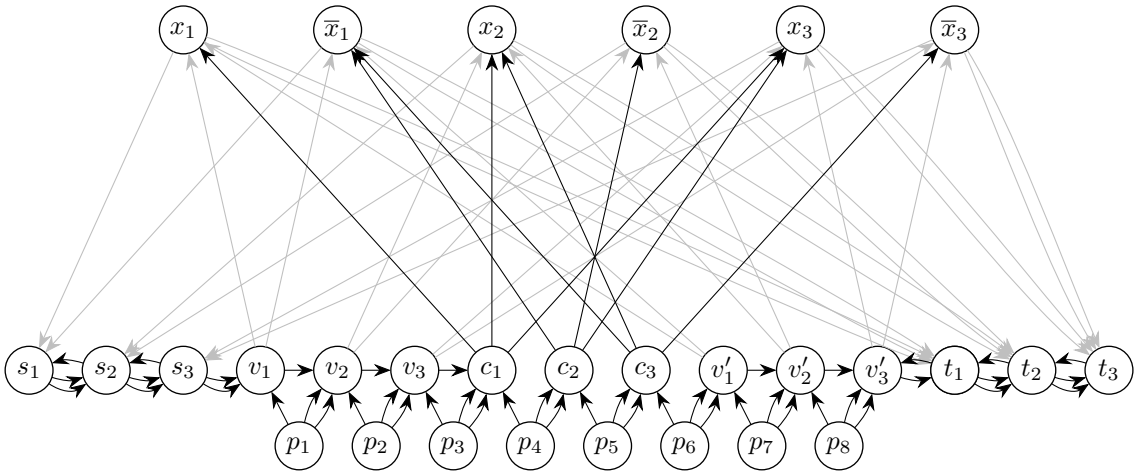


Figure 2: The graph constructed in the proof of Theorem 9 for the CNF formula $(x_1 \vee x_2 \vee x_3) \wedge (\overline{x}_1 \vee \overline{x}_2 \vee x_3) \wedge (\overline{x}_1 \vee x_2 \vee \overline{x}_3)$.

Let D contain the vertices x_i and $\overline{x}_i$ for each literal (see the top row of vertices in Figure 2), two vertices v_i and v'_i for each variable x_i, and a vertex c_j corresponding to the j^{th} clause for $j \in \{1, \ldots, n\}$. Each vertex c_j has 3 outgoing arcs to the vertices x_i or $\overline{x}_i$ corresponding to the literals contained in c_j. For each $i \in \{1 \ldots, n\}$, both vertices v_i and v'_i have two outgoing arcs to the vertices x_i and $\overline{x}_i$. Let D contain two other vertices denoted by s_i and t_i for $j \in \{1, \ldots, n\}$, and from each vertex x_i and $\overline{x}_i$ add an arc to s_i and two parallel arcs to t_i. Consider the vertices $s_1, \ldots, s_n, v_1, \ldots, v_n, c_1, \ldots, c_n, v'_1, \ldots, v'_n, t_1, \ldots, t_n$ in this order, see Figure 2. Let D contain an arc from the vertices v_i and v'_i to the succeeding vertex, and from each vertex s_i and t_i two parallel outgoing arcs to the succeeding and one outgoing arc to the preceding vertex. Moreover, let D contain an additional vertex between every two adjacent vertices of the

sequence $v_1, \ldots, v_n, c_1, \ldots, c_n, v'_1, \ldots, v'_n$, and let $p_1, \ldots, p_{3n-1}$ denote these newly added vertices. For each $k \in \{1, \ldots, 3n-1\}$, let p_k have two parallel outgoing arcs to the succeeding vertex and one outgoing arc to the preceding vertex. Let the bounds be $f(s_1) = g(s_1) = 0$ and $f(t_n) = g(t_n) = 1$, and for each vertex $v \neq \{s_1, t_n\}$, $f(v) = 1$ and $g(v) = \delta(v) - 2$. We show that for a given CNF formula the 3-XSAT-3 problem is satisfiable if and only if the (f, g)-FAS problem defined on D is solvable.

If the instance of the 3-XSAT-3 problem is satisfiable, then consider the following order of the vertices of D: First, list the vertices $s_1, \ldots, s_n, v_1, \ldots, v_n, c_1, \ldots, c_n, v'_1, \ldots, v'_n, t_1, \ldots, t_n$ in this order, and put each additional vertex p_k between its two neighbors, see Figure 2. Then put the vertices of the true literals right after the vertex s_n and the vertices of the false literals right before the vertex t_1. By the construction, it is easy to see that the resulting order is a feasible solution to the (f, g)-FAS problem.

Conversely, let the order σ be a feasible solution to the (f, g)-FAS problem. Notice that the additional vertices $p_1, \ldots, p_{3n-1}$ ensure that the vertices $v_1, \ldots, v_n, c_1, \ldots, c_n, v'_1, \ldots, v'_n$ appear in this order. Furthermore, the vertices $s_1, \ldots, s_n$ must precede this sequence and the vertices $t_1, \ldots, t_n$ must succeed this sequence, because of their outgoing parallel arcs. Therefore, the vertices $s_1, \ldots, s_n, v_1, \ldots, v_n, c_1, \ldots, c_n$, $v'_1, \ldots, v'_n, t_1, \ldots, t_n$ must appear in this order. The vertices v_i and v'_i corresponding to the variable x_i have exactly 3 outgoing arcs, and one of these arcs is going to the right. Therefore, for both v_i and v'_i, one of the remaining two arcs going to x_i and to $\overline{x}_i$ must point to the left and the other one to the right by the bounds f and g. This means that, for each variable, one of the vertices x_i and $\overline{x}_i$ precedes the vertex v_i and the other one succeeds the vertex v'_i. Because of the fixed order of the vertices $v_1, \ldots, v_n, c_1, \ldots, c_n$, $v'_1, \ldots, v'_n$, this implies that, for each variable, one of the two literals x_i and $\overline{x}_i$ must be placed before $c_1, \ldots, c_n$ and the other one after them. Set the variable x_i to true if the literal x_i precedes the vertices $c_1, \ldots, c_n$, and to false otherwise, hence exactly the literals preceding the vertices $c_1, \ldots, c_n$ are true. This is a solution to the instance of the 3-XSAT-3 problem, because each vertex c_j has 3 outgoing arcs to the vertices corresponding to the literals in c_j, and exactly one of these precedes the vertex c_j by the bounds f and g — and hence it also precedes $c_1, \ldots, c_n$. Therefore, exactly one literal is true in each clause. □

For the digraph D, the (f, g)-FAS problem defined in the proof is solvable if and only if the (f, g)-FAS problem with bounds $f(s_1) = g(s_1) = 0$ and $f(v) = g(v) = 1$ for each vertex $v \neq s$ is solvable. Therefore, the proof also shows that the case when $f \equiv g$ is NP-complete.

Corollary 10 *The (f, g)-FAS problem with $f \equiv g$ is NP-complete. The problem is NP-complete even when the $f \equiv g$ bound is 0 on one vertex, and 1 on all other vertices.*

Moreover, the problem is NP-complete for all parameters $a \geq 1$ and $b \geq 2$, so essentially in every case with stricter bounds than in the s-t numbering problem. This follows by a reduction from the $a = 1$ and $b = 2$ case by adding $a - 1$ new arcs from v to s and $b - 2$ new arcs from v to t for each vertex $v \neq \{s, t\}$.

Corollary 11 *For all parameters $a \geq 1$ and $b \geq 2$, the (f, g)-FAS problem with bounds $f(v) = a$, $g(v) = \delta(v) - b$ on each vertex, except the first and the last ones, is NP-complete.*

3 Special arc-partition problems

This section considers arc-partition problems in which the goal is to decide whether a digraph can be partitioned into a subgraph with special properties and an acyclic subgraph. For example, partitioning into an in-branching and an acyclic subgraph can be solved in polynomial time by using the algorithm from Section 2.1 for the $(-\infty, g)$-FAS problem with upper bound $g \equiv 1$. Similar arc-partition problems are discussed in [2]. They proved that it is NP-complete to decide whether a digraph can be partitioned into a directed cycle and an acyclic subgraph, or into a directed cycle factor and an acyclic subgraph.

3.1 In-branching and acyclic subgraph

Theorem 12 *A digraph $D = (V, A)$ can be partitioned into an in-branching $B \subseteq A$ and an acyclic subgraph if and only if there exists no induced subgraph $D' = (V', A')$ of D in which the out-degree of each vertex $v \in V'$ is at least 2.*

PROOF: The arc-partition problem is equivalent to the $(-\infty, g; \sum w)$-FAS problem with upper bound $g \equiv 1$, so the characterization follows from Theorem 5. □

Furthermore, a similar characterization holds for the case when some vertices are required to be roots in the in-branching.

Theorem 13 *Given a digraph $D = (V, A)$ and a subset $X \subseteq V$ of the vertices, it can be decided in polynomial time whether the digraph can be partitioned into an acyclic subgraph and an in-branching $B \subseteq A$ in which the vertices in X are roots. Such a partition exists if and only if there exists no induced subgraph $D' = (V', A')$ of D in which the out-degree of each vertex $v \in X$ is at least one and the out-degree of each vertex $v \in V' \setminus X$ is at least 2.*

It is important to note that the in-branching may contain roots other than the vertices in X. Therefore, this theorem is not applicable to partition a digraph into an in-arborescence and an acyclic subgraph. The complexity of this problem remains open. However, partitioning into an in-arborescence and a *spanning* acyclic subgraph is known to be NP-hard [2]. Next, we show that partitioning to an acyclic subgraph and a minimum-size in-branching or a minimum-cost in-arborescence are NP-hard problems.

Theorem 14 *It is NP-hard to find a minimum-size in-branching in a digraph whose complement is acyclic.*

PROOF: The proof is by reduction from the vertex cover problem [10]. Let $G = (V, E)$ be the graph for which we want to find a minimum vertex cover. Construct the digraph $D = (V \cup V', A)$ as follows. Let V' be a copy of V, and let $v' \in V'$ denote the copy of $v \in V$. Add an arc vv' for each vertex $v \in V$ to D. Moreover, let D contain two arcs $u'v$ and $v'u$ for each edge $uv \in E$, where $u' \in V'$ is the copy of $u \in V$. We show that G has a vertex cover of size at most k if and only if D contains an in-branching of size at most k whose complement is acyclic.

If G has a vertex cover $\{v_1, \ldots, v_\ell\} \subseteq V$ with $\ell \leq k$, then the arcs $v_1v'_1, \ldots, v_\ell v'_\ell \in A$ form an in-branching that covers all directed cycles in D.

Conversely, there exists an in-branching of size at most k that covers all directed cycles in D. Notice, that if a directed cycle contains the arc $u'v$, then it also contains the arc uu', because it is the only arc entering u'. Therefore, we can replace each arc $u'v$ from V' to V in the in-branching with uu', and the arcs $v_1v'_1, \ldots, v_\ell v'_\ell$ obtained this way cover all directed cycles in D. This arc set must contain uu' or vv' for each edge $uv \in E$, because D contains a directed cycle $uu'vv'$ for each edge $uv \in E$. Therefore, the vertices $v_1, \ldots, v_\ell \in V$ form a vertex cover in G with size at most k. □

The NP-hardness of partitioning into a minimum-cost in-arborescence and acyclic subgraph follows by a reduction from partitioning into a minimum-size in-branching and acyclic subgraph. The main idea is that we add a new vertex s to the graph with an outgoing arc to all other vertices. Let the cost function be 0 on these new arcs, and 1 on the rest of the arcs. This graph contains an appropriate in-arborescence of cost at most k if and only if the original graph contains an appropriate in-branching of cost at most k. Hence we get the following.

Theorem 15 *Let us given a digraph $D = (V, A)$ with a 0-1 cost function on the arc-set. It is NP-hard to find a minimum-cost in-arborescence whose complement is acyclic.*

3.2 Matching and acyclic subgraph

It is a similar problem to decide whether a digraph can be partitioned into a matching and an acyclic subgraph. Motivated by the solvability of partitioning into an in-branching and an acyclic subgraph, it is natural to ask whether the problem is solvable in the case of matchings. In what follows, we show that this problem is NP-complete.

Theorem 16 *It is NP-complete to decide whether a digraph can be partitioned into a matching and an acyclic subgraph.*

PROOF: *Sketch of the proof.* The problem is clearly in NP. We only describe the construction of the reduction from the NAE-3-SAT problem [12]. In the NAE-3-SAT problem the goal is to decide whether a CNF formula in which each clause contains exactly 3 literals can be satisfied such that each clause has at least one false literal in it. For a given CNF formula, construct the digraph D as follows: For each clause c_j, let D contain a gadget on 9 vertices, denoted by u_k^j, c_k^j and $\overline{c}_k^j$ for $k = 1, 2, 3$. The vertex c_k^j corresponds to the k^{th} literal of the clause c_j, and the vertex $\overline{c}_k^j$ corresponds to its negation. The gadget contains a directed cycle $c_1^j u_1^j c_2^j u_2^j c_3^j u_3^j$ and a directed cycle $\overline{c}_1^j u_3^j \overline{c}_3^j u_2^j \overline{c}_2^j u_1^j$. Figure 3 illustrates the construction for the clause c_j. Moreover, for each variable x_i, let D contain a gadget on the vertices v_ℓ^i for $\ell = 1, \ldots, 5$ with an arc $v_1^i v_2^i$ and an arc $v_2^i v_1^i$, and a directed cycle $v_2^i v_3^i v_4^i v_5^i$. The gadgets belonging to the clauses and to the literals are connected to each other as follows:

If the k^{th} literal of the clause c_j is x_i, then extend the gadget belonging to x_i with two vertices denoted by y_k^j and z_k^j and connect the vertices c_k^j and $\overline{c}_k^j$ from the gadget belonging to c_j with a path $v_5^i \overline{c}_k^j z_k^j v_4^i y_k^j c_k^j v_3^i$. Similarly, if the k^{th} literal of the clause c_j is $\overline{x}_i$, then extend the gadget belonging to x_i with two vertices denoted by y_k^j and z_k^j and connect the vertices c_k^j and $\overline{c}_k^j$ from the gadget belonging to c_j with a path $v_5^i c_k^j z_k^j v_4^i y_k^j \overline{c}_k^j v_3^i$. Figure 4 illustrates the gadget belonging to the variable x_i, and its two possible extensions with respect to clause c_j containing x_i.

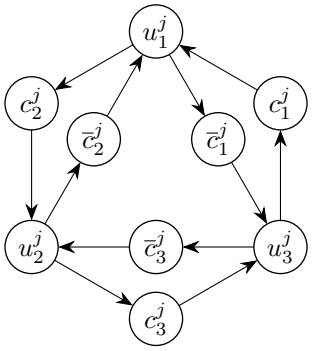


Figure 3: The gadget belonging to the clause c_j.

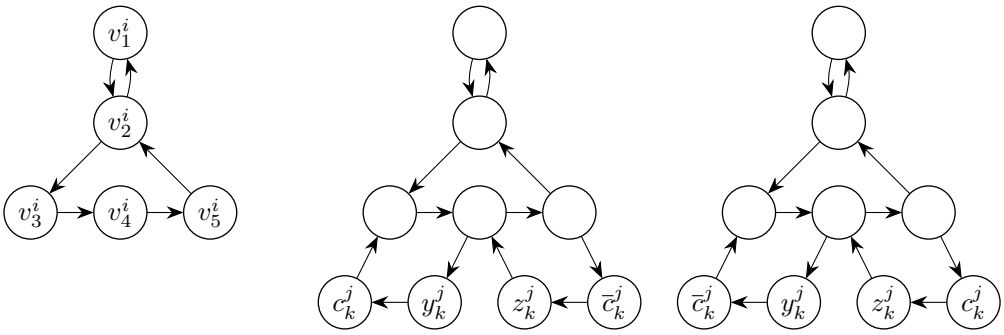


Figure 4: The gadget belonging to the variable x_i and its extensions if the k^{th} literal in the clause c_j is x_i or if the k^{th} literal in the clause c_j is $\overline{x}_i$. The gadget has an extension for each clause containing the literals x_i or $\overline{x}_i$. The vertices with labels c_k^j and $\overline{c}_k^j$ are identical to those with the same label in Figure 3.

It can be proved that the NAE-3-SAT problem is solvable if and only if the digraph D constructed above can be partitioned into a matching and an acyclic subgraph. The variable x_i is true in this solution if and only if the matching contains the arc $v_3^i v_4^i$ in the gadget belonging to the variable x_i. □

By a simple modification of this construction, one gets that the problem is also NP-complete when the matching is required to be perfect.

4 Open questions

The most interesting open question is the complexity of partitioning into an in-arborescence and an acyclic subgraph. We proved in Section 2 that the similar problem of partitioning into an in-branching and an acyclic subgraph is polynomial-time solvable.

It is proved in [2] that decomposing into a directed cycle factor and an acyclic subgraph is NP-complete, however, the complexity remains open if the directed cycles are only required to be disjoint but they not necessarily cover all vertices. There are a few other open problems mentioned in [2], for example, covering all odd directed cycles with a perfect matching, or partitioning into a perfect matching and a subgraph containing an in-arborescence.

Finally, we mention two natural generalizations of the $(-\infty, g; h)$-ordering problem. Motivated by the position-based scheduling problem [9], let us given a cost for each item-place pair, and search for a minimum-cost $(-\infty, g; h)$-order. Another natural generalization is if, instead of a linear order, we want to find a two dimensional arrangement. In this case, the items preceding v can be defined as the items placed left-down from v.

References

[1] J. Bang-Jensen. Edge-disjoint in- and out-branchings in tournaments and related path problems. *Journal of Combinatorial Theory, Series B*, 51(1):1–23, 1991.

[2] J. Bang-Jensen, S. Bessy, D. Gonçalves, and L. Picasarri-Arrieta. Complexity of some arc-partition problems for digraphs. *Theoretical Computer Science*, 928:167–182, 2022.

[3] J. Bang-Jensen and C. J. Casselgren. Restricted cycle factors and arc-decompositions of digraphs. *Discrete Applied Mathematics*, 193:80–93, 2015.

[4] J. Bang-Jensen, G. Gutin, and A. Yeo. Arc-disjoint strong spanning subdigraphs of semicomplete compositions. *Journal of Graph Theory*, 95(2):267–289, 2020.

[5] J. C. Bermond and V. Faber. Decomposition of the complete directed graph into k-circuits. *Journal of Combinatorial Theory, Series B*, 21(2):146–155, 1976.

[6] T. Biedl, F. J. Brandenburg, and X. Deng. On the complexity of crossings in permutations. *Discrete Mathematics*, 309(7):1813–1823, 2009.

[7] J. Cheriyan and J. H. Reif. Directed s-t numberings, rubber bands, and testing digraph k-vertex connectivity. *Combinatorica*, 14(4):435–451, 1994.

[8] J. Edmonds. Edge-disjoint branchings. *Combinatorial algorithms*, pages 91–96, 1973.

[9] M. Horváth and T. Kis. Polyhedral results for position-based scheduling of chains on a single machine. *Annals of Operations Research*, 284(1):283–322, 2020.

[10] R. M. Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.

[11] J. Opatrny. Total ordering problem. *SIAM Journal on Computing*, 8(1):111–114, 1979.

[12] S. Porschen, T. Schmidt, E. Speckenmeyer, and A. Wotzlaw. XSAT and NAE-SAT of linear CNF classes. *Discrete Applied Mathematics*, 167:1–14, 2014.

Data Augmentation Does Not Necessarily Beat a Smart Algorithm

Krisztian Buza[1]

Institute Jozef Stefan
Artificial Intelligence Laboratory
Jamova 39, 1000 Ljubljana, Slovenia

BioIntelligence Group
Department of Mathematics-Informatics
Sapientia Hungarian University of Transylvania
Targu Mures, Romania

buza@biointelligence.hu

Abstract: According to the "widely acknowledged truth", more training data beats algorithmic improvements in machine learning tasks. We challenge this "widely acknowledged truth" in context of data augmentation of images and recognition tasks related to images. Our observations show that real training data may be much more valuable than augmented (i.e., artificially generated) data and – most importantly – the advantage of a sophisticated algorithm relative to a simple algorithm may not be easily compensated by data augmentation.

Keywords: dynamic programming, data augmentation, machine learning, dynamic image warping

1 Introduction

State-of-the art solutions of various recognition tasks, ranging from handwriting recognition and signature verification over biometric user identification (e.g. based on the dynamic of typing) to speech recognition and image analysis tasks, are based on machine learning. Especially deep neural networks became extraordinarily popular in the last decade. Spectacular results include the detection of skin cancer [1] and retinal disease [2], "mastering the game of Go" [3], as well as recognition tasks relevant for the automotive industry [4]. Nevertheless, training deep neural networks requires a very large set of training data which is usually expensive and difficult to collect, if not impossible. For example, in case of rare diseases it may not be possible to obtain data from millions of patients. In case of biometric authentication systems, when a new user signs up to the system, the user may be asked to provide her biometric (such as handwriting or typing dynamics) a *few* times, but not thousands or millions of times.

In order to alleviate the aforementioned issues related to the collection of very large datasets, one of the most popular techniques is to generate new instances from existing instances by the (minor) modification or combination of existing instances. For example, in case of image recognition tasks, images may be shifted, elongated, resized or rotated by a few degrees, see also Fig. 2.

While data augmentation is further justified by observations showing that the prediction accuracy of machine learning techniques improves with increasing amount of training data, the actual data augmentation techniques may be somewhat ad hoc and understudied from the point of view of theory.

[1]This work was supported by the European Union through enRichMyData EU HE project under grant agreement No 101070284.

Furthermore, if sufficient training data is provided, simple algorithms have been shown to work surprisingly well in many domains[1], see e.g. nearest neighbor algorithms in time series classification [5], modified linear regression in case of drug-target interaction prediction [6] or simple classifiers in case of IT ticket text classification [7]. Moreover, Schnoebelen points out that "the widely acknowledged truth is that throwing more training data into the mix beats work on algorithms."[2]

In this paper, we will examine to which extent this "widely acknowledged truth" applies to data augmentation, one of the most prominent techniques used to improve the performance of deep neural networks. In particular, we consider an an elastic distance measure, *dynamic image warping* (DIW) which is a recent extension of dynamic time warping (DTW) for images. We compare DIW to simple distance measures, such as Euclidean and Manhattan distance. In our experiment on images of handwritten digits, DIW outperforms Euclidean and Manhattan distance even in case of data augmentation which indicates that the advantage of the more sophisticated algorithm may not be easily compensated by data augmentation.

The reminder of the paper is organized as follows: Section 2 presents dynamic image warping, while Section 3 describes our empirical observations as well as the lessons we learned from our experiments.

2 Dynamic Image Warping

Dynamic Time Warping (DTW) is an elastic distance measure for time series [8]. While comparing two time series, DTW allows for shifts and elongations. This way, DTW takes into account that, in real-world data, the same pattern is not likely to be repeated in the exactly same way. DTW is based on the paradigm of dynamic programming. When implementing DTW calculations, the entries of a matrix are filled according to a recursive rule. For more details on DTW, we refer to [9].

Dynamic Image Warping (DIW) is a recent extension of dynamic time warping (DTW) for images [10]. A digital image is a matrix of intensity values. For simplicity, we only consider grayscale images in this paper, thus, each pixel corresponds to a single intensity value. However, we note that the generalisation for color (RGB) images is straightforward. As a first step of DIW, we consider the intensity matrix of the image as a sequence of rows (or columns, respectively).

In order to compare two images, DIW compares two *sequences of sequences*. We note that in case of time series, DTW compares two *sequences of numbers* which is the major difference between DIW and DTW.

When calculating DIW, in principle, we follow the same steps as in case of DTW. The only difference between DIW and DTW is the following: while at some steps of DTW, the difference of two numbers have to be calculated, at the corresponding step in DIW, we have to calculate the distance of two sequences. In order to calculate the distance of those two sequences, we use DTW. In other words: DIW is nothing else but DTW for a sequence of sequences using DTW as inner distance.

We note that the role of columns and rows is interchangeable in case of images, therefore, when implementing our experiments, we actually calculate two DIW distances: in case of the first one, each image is considered as a sequence of rows, whereas in case of the second distance, each image is considered as a sequence of columns.

Considering an image with $N \times N$ pixels, DIW has a complexity of $\mathcal{O}(N^4)$. For this reason, we implemented DIW in Cython [11] in order to combine the efficiency of C with rapid prototyping allowed by Python. For more details see:

https://github.com/kr7/diw/blob/main/DIW.ipynb

[1] See also https://anand.typepad.com/datawocky/2008/04/data-versus-alg.html for a related discussion.
[2] https://www.datasciencecentral.com/more-data-beats-better-algorithms-by-tyler-schnoebelen/

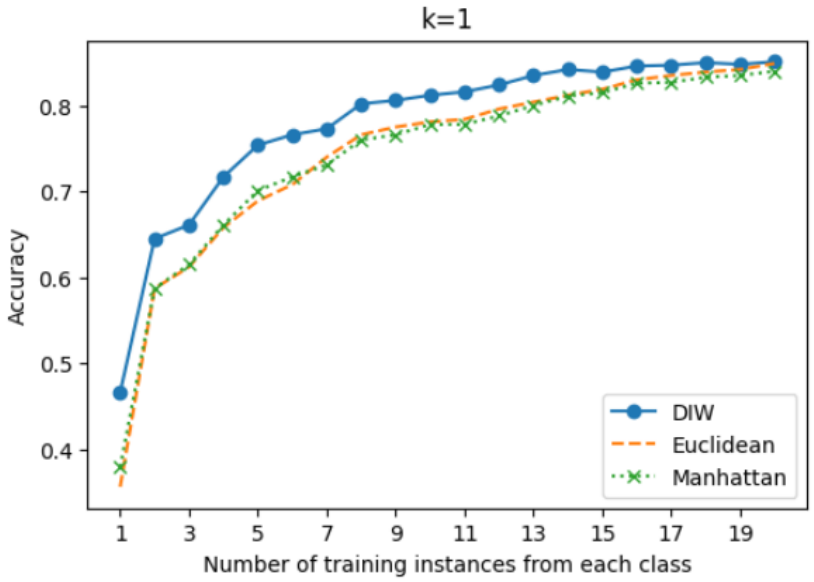


Figure 1: Classification Accuracy as Function of the Number of Training Instances

3 Experiments

Next, we present our empirical observations. In the first experiment, the classification accuracy as function of the number of training instances is studied, whereas in the second experiment, we examine the effect of data augmentation.

3.1 Classification Accuracy as Function of the Number of Training Instances

We performed experiments in the context of the recognition of handwritten digits. This is a classification task with 10 classes, where each of the classes corresponds to one of the digits '0', '1', '2', ..., '9', see also the left column of Fig. 2 for examples of images from our dataset. In our experiment, we aimed to recognize the handwritten digit using a 1-nearest neighbor classifier using either (i) DIW, or (ii) Euclidean distance or (iii) Manhattan distance to determine the nearest neighbor.

Fig. 1 shows the classification accuracy as function of the number of training instances. For example, in the case when five instances are used from each class, the training data contained five images showing a '0', another five images showing a '1', etc., thus the total size of the training data was $5 \times 10 = 50$. As one can see, the classification accuracy increases with increasing size of training data. While DIW outperforms the two other distance measures in case of few instances, the difference between the performance of the three approaches gradually decreases. When using $20 \times 10 = 200$ training instances, the classification accuracy of more than 80% is reached.

This experiment can be reproduced by running the Google colab notebook available at

https://github.com/kr7/diw/blob/main/DIW.ipynb ,

we refer to this code for further details (such as the URL of the dataset, training and test splits).

3.2 Data Augmentation

As Fig. 1 shows, in the previous experiment, compared with the case of using a single training instance per class, we observed substantial improvement in terms of classification accuracy when using more training

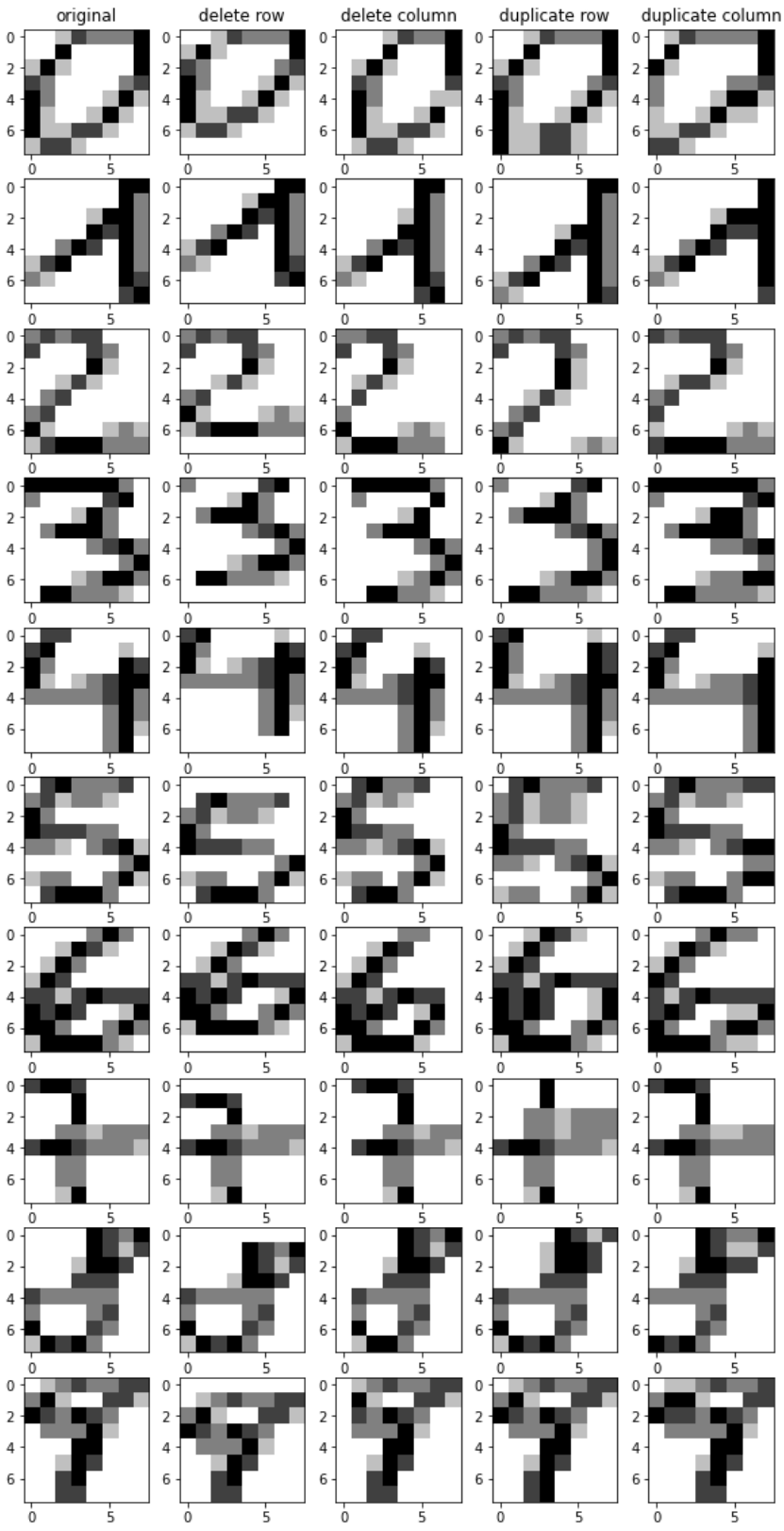


Figure 2: Data augmentation techniques used in our experiment.

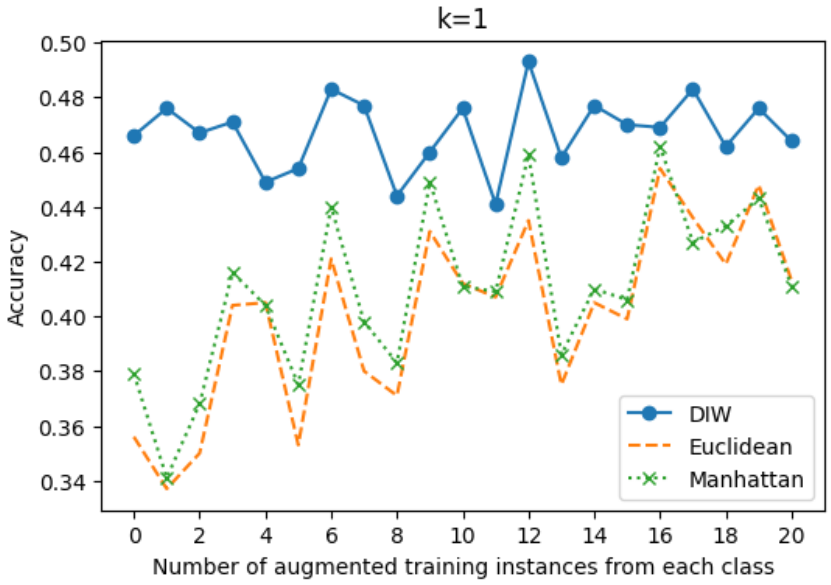


Figure 3: Classification Accuracy as Function of the Number of Training Instances

data. Next, we examine if the same improvement can be achieved using simple data augmentation techniques.

Fig. 2 shows the data augmentation techniques used in our experiment. The leftmost column shows the original image, while subsequent columns show the image after (i) deleting a randomly selected row, (ii) deleting a randomly selected column, (iii) duplication of a randomly selected row and (iv) duplication of a randomly selected column.

In this experiment, we only consider a single training instance per class. For each training instance, we create t augmented instances using the aforementioned augmentation techniques. These t augmented instances are added to the training set and the performance of the 1-nearest neighbor classifier is measured on the test set.

Fig. 3 shows the accuracy of the classifier as function of t in cases when (i) DIW, (ii) Euclidean distance and (iii) Manhattan distance was used to determine the nearest neighbor. This experiment can be reproduced by running the Google colab notebook available at

https://github.com/kr7/diw/blob/main/DIW-augmentation.ipynb .

Please see this code for further details.

Based on Fig. 3, we can make the following observations:

1. Data augmentation does not necessarily improve the performance. Is seems that data augmentation may introduce noise, although the accuracy has an increasing trend in case Euclidean and Manhattan distances.

2. More importantly, even in case of augmented data, the more sophisticated DIW algorithm beats simple distance measures which indicates that the advantage of the more sophisticated algorithm may not be easily compensated by data augmentation.

3. Last, but not least we have to note that, using data augmentation, we were not able to achieve an accuracy that is comparable with the case of using real observations as training data in Section 3.1.

An inherent limitation of our work is that we performed experiments only on a relatively small dataset from the domain of handwriting recognition which may be considered simple compared to industrial

domains, such as self-driving vehicles. While we clearly admit this limitation, we note that our setting aimed to simulate more complex domains: imagine, for example a 1 MP color (RGB) image. This image corresponds to a point in a 3 million dimensional vector space. As high-dimensional data spaces tend to be increasingly sparse, our setting, in which we only considered a single training instance when augmenting the data, aims to simulate such a sparsity.

References

[1] A. Esteva, et al., Dermatologist-level classification of skin cancer with deep neural networks, *nature* **542.7639** (2017)

[2] J. De Fauw, et al., Clinically applicable deep learning for diagnosis and referral in retinal disease, *Nature medicine* **24.9** (2018)

[3] D. Silver, et al., Mastering the game of Go with deep neural networks and tree search, *nature* **529.7587** (2016)

[4] A. Luckow, et al., Deep learning in the automotive industry: Applications and tools, *IEEE International Conference on Big Data* (2016)

[5] X. Xi, et al., Fast time series classification using numerosity reduction, *Proceedings of the 23rd international conference on Machine learning* (2006)

[6] K. Buza, L. Peška, J. Koller, Modified linear regression predicts drug-target interactions accurately, *PloS one* **15.4** (2020)

[7] A. Revina, K. Buza, V. G. Meister, IT ticket classification: the simpler, the better, *IEEE Access* **8** (2020)

[8] H. Sakoe, S. Chiba, Dynamic programming algorithm optimization for spoken word recognition *IEEE transactions on acoustics, speech, and signal processing* **26.1** (1978)

[9] K. Buza, Time series classification and its applications *Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics* (2018)

[10] K. Buza, M. Antal, An Extension of Dynamic Time Warping for Images: Dynamic Image Warping, *14th Joint Conference on Mathematics and Computer Science* (2022)

[11] S. Behnel, et al, Cython: The best of both worlds, *Computing in Science & Engineering* **13.2** (2010)

Approximation Algorithms for Matroidal and Cardinal Generalizations of Stable Matching*

Gergely Csáji[1]
Department of Operations Research
Eötvös Loránd University
Budapest, Pázmány Péter promenade 1/A, Hungary
csaji.gergely@student.elte.hu

Tamás Király[2]
ELKH-ELTE Egerváry Research Group
Eötvös Loránd University
Budapest, Pázmány Péter promenade 1/A, Hungary
tamas.kiraly@ttk.elte.hu

Yu Yokoi[3]
Principles of Informatics Research Division
National Institute of Informatics
Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan
yokoi@nii.ac.jp

Abstract: The Stable Marriage problem (SM), solved by the famous deferred acceptance algorithm of Gale and Shapley (GS), has many natural generalizations. If we allow ties in preferences, then the problem of finding a maximum solution becomes NP-hard, and the best known approximation ratio is 1.5 (McDermid ICALP 2009, Paluch WAOA 2011, Z. Király MATCH-UP 2012), achievable by running GS on a cleverly constructed modified instance. Another elegant generalization of SM is the matroid kernel problem introduced by Fleiner (IPCO 2001), which is solvable in polynomial time using an abstract matroidal version of GS. Our main result is a simple 1.5-approximation algorithm for the matroid kernel problem with ties. We also show that the algorithm works for several other versions of stability defined for cardinal preferences, by appropriately modifying the instance on which GS is executed. The latter results are new even for the stable marriage setting.

Keywords: stable matching, matroid kernel, approximation, deferred acceptance

1 Introduction

The deferred acceptance algorithm of Gale and Shapley [8] is a quintessential example of a simple combinatorial algorithm that has wide-ranging applications, in such diverse areas as healtchcare labor markets, kidney exchange planning, project allocations, and school choice mechanisms. The original stable marriage problem solved by the Gale–Shapley algorithm has been generalized in many directions, and the mathematical research in the area is still thriving, 60 years after the original paper.

The aim of the present paper is to bring together two directions in which the problem has been extended. One is the design of approximation algorithms for finding a maximum stable matching when ties are allowed in the preference lists. The other is the generalization of the stable matching problem to matroid intersection, in particular, the matroid kernel problem introduced and solved by Fleiner [6] using an abstract version of the Gale–Shapley algorithm.

*A preliminary version of this paper will appear in the proceedings of the sixth SIAM Symposium on Simplicity of Algorithms (SOSA), 2023. The authors would like to thank Tamás Fleiner, Zsuzsanna Jankó, and Ildikó Schlotter for fruitful discussions.

[1]Research was supported by the Lendület Programme of the Hungarian Academy of Sciences – grant number LP2021-1/2021, by the Hungarian National Research, Development and Innovation Office – NKFIH, grant number K143858.

[2]Research was supported by the Lendület Programme of the Hungarian Academy of Sciences – grant number LP2021-1/2021, by the Hungarian National Research, Development and Innovation Office – NKFIH, grant numbers TKP2020-NKA-06 and K143858.

[3]Research was supported by the JST PRESTO Grant Number JPMJPR212B.

We show that the best known approximation ratio of 1.5 for the stable marriage problem with ties [16] can also be achieved for the matroid kernel problem with ties. Furthermore, we can go beyond ordinal preferences with ties, and achieve the same approximation ratio for problems with cardinal preferences and various near-stability requirements. In all of these cases, our algorithms are simple variants of the Gale–Shapley algorithm applied to carefully constructed modified instances of the problem. The key observation that enables the extension of the proof to matroid kernels is an exchange property for ordered matroids that may be of independent interest.

1.1 Basic definitions

Stable marriage with ties.

A *weak order* on a ground set S is an ordering of S that may contain ties; in other words, it is a partial order where being incomparable is a transitive relation. We use $\succsim$ to denote a weak order; if several weak orders are given, we use indices to distinguish them. The notation $x \sim y$ means that x is tied with y.

In the *stable marriage problem with ties and incomplete lists* (SMTI), we are given a bipartite graph $G = (U, W; E)$, and a weak order $\succsim_v$ on $\delta_G(v)$ for every vertex $v \in U \cup W$, where $\delta_G(v)$ denotes the set of edges incident to v in G. Given a matching N in G, an edge $e = uw \in E \setminus N$ is a *blocking edge* for N if the following two conditions hold:

- $\delta_G(u) \cap N = \emptyset$ or $e \succ_u N(u)$,
- $\delta_G(w) \cap N = \emptyset$ or $e \succ_w N(w)$,

where $N(u)$ denotes the edge of N incident to u if it exists. The matching N is *stable* if no edge blocks it. The Max-SMTI problem is to find a stable matching of maximum size. The problem is NP-hard [10], and the best known polynomial-time approximation algorithm has an approximation ratio of 1.5 [16]. It is also known that no approximation ratio better than 4/3 is achievable assuming the Unique Games Conjecture [21].

If no ties are allowed in the preference orders, then we obtain the standard stable marriage problem, where all stable matchings have the same size, and the Gale–Shapley algorithm finds one efficiently.

Matroid kernels.

A natural way to generalize the stable marriage problem and the Max-SMTI problem is to allow agents to have multiple partners, but have some restrictions on the possible sets of partners. There are several models in this vein: the hospitals-residents problem, the college admissions problem with common quotas, classified stable matchings etc. (see subsection 1.3 for more details).

From a theoretical point of view, a particularly elegant generalization is the matroid kernel problem defined by Fleiner [6]. We assume familiarity with the fundamental notions of matroid theory (independence, bases, fundamental circuits). We use the notation and terminology of [20, Chapter 39] for matroids, unless otherwise stated. A *weakly ordered matroid* is a triple $(S, \mathcal{I}, \succsim)$ where S is the set of elements, $\mathcal{I}$ is the family of independent sets of the matroid, and $\succsim$ is a weak order on S. Let $M_1 = (S, \mathcal{I}_1, \succsim_1)$ and $M_2 = (S, \mathcal{I}_2, \succsim_2)$ be weakly ordered matroids on the same ground set S. A common independent set $X \in \mathcal{I}_1 \cap \mathcal{I}_2$ is an (M_1, M_2)-*kernel* if for every $y \in S \setminus X$ there exists $i \in \{1, 2\}$ such that $X + y \notin \mathcal{I}_i$ and $x \succsim_i y$ for every $x \in X$ for which $X - x + y \in \mathcal{I}_i$. If there is an element $y \in S \setminus X$ for which this does not hold, then we say that y *blocks* X. The Max-Kernel problem is to find an (M_1, M_2)-kernel of maximum size.

If M_1 and M_2 are partition matroids, then the Max-Kernel problem is equivalent to Max-SMTI. Indeed, we can construct a bipartite graph by considering the partition classes of the two matroids as vertices, and the elements of S as edges, whose two endpoints are the vertices corresponding to the two partition classes containing that element. There is a one-to-one correspondence between stable matchings of this bipartite graph and (M_1, M_2)-kernels.

Fleiner [6, 7] considered the matroid kernel problem without ties, i.e., $\succsim_1$ and $\succsim_2$ are linear orders. He showed that matroid kernels always exist and have the same size (in fact, they have the same span in both matroids). He also gave a matroidal version of the Gale–Shapley algorithm that finds a matroid kernel efficiently. In case of weak orders, kernels may have different sizes, and it is NP-hard to find a largest one, since this problem is a generalization of Max-SMTI.

Cardinal preferences and near-stability.

One way to define a weak order on a ground set S is to assign a preference value $p(x)$ to each $x \in S$, with larger value being better. Without loss of generality, we will assume that each $p(x)$ is nonnegative, and we also define $p(\emptyset) = 0$. Such cardinal preferences also allow the definition of various versions of near-stability. Intuitively, the blocking of a solution may require some quantifiable effort from the blocking agents, so we may say that an element does not block unless the improvement is at least some fixed positive Δ.

We define near-stability in the context of matroid kernels; the definitions carry over naturally to the special case of stable marriage (for the latter, similar definitions appeared in the literature under various names [1, 4, 19]). For $i \in \{1, 2\}$, let $M_i = (S, \mathcal{I}_i)$ be matroids, and for each element $x \in S$ let $p_i(x) \geq 0$ be its value in the matroid M_i. The values p_i define a weak order $\succsim_i$ on S by $x \succsim_i y \Leftrightarrow p_i(x) \geq p_i(y)$. As mentioned above, we assume that $p_1(\emptyset) = p_2(\emptyset) = 0$.

Let $\Delta > 0$. Given a common independent set X, an element $y \in S \setminus X$ is said to be Δ-*min-blocking* for X if there exist $x_1 \in X \cup \{\emptyset\}$ and $x_2 \in X \cup \{\emptyset\}$ such that

- $x_i = \emptyset$ if $X + y \in \mathcal{I}_i$, otherwise $X - x_i + y \in \mathcal{I}_i$, for $i \in \{1, 2\}$,
- $\min\{\, p_1(y) - p_1(x_1),\, p_2(y) - p_2(x_2) \,\} \geq \Delta$.

Informally, an element Δ-min-blocks X if we can achieve an improvement of Δ in both matroids, by adding it to X or by exchanging it with an element of X. The set X is Δ-*min-stable* if there is no Δ-min-blocking element. If all p_i values are positive and Δ is small enough, then Δ-min-stability is equivalent to being an (M_1, M_2)-kernel. Note also that in the stable marriage setting, it would make sense to have a different threshold (i.e., a different Δ value) for each agent. However, by rescaling the preference values appropriately, we can assume that all these thresholds are the same.

We will also consider two other versions of near-stability, for which 1.5-approximation can be achieved using a slightly more complicated construction. These are presented in Section 4.

1.2 Our contribution

The key tool for generalizing the 1.5-approximation to matroid kernels is a result on the existence of a perfect matching of certain types of exchangeable pairs in a matroid. This is presented as Theorem 3 in Section 2.

Using Theorem 3, we show in Section 3 that there is a 1.5-approximation algorithm for Max-Kernel, which consists of three steps: (1) constructing an instance of the matroid kernel problem without ties on the ground set obtained by replacing each element by 3 parallel elements, (2) running Fleiner's algorithm on the new instance, and (3) projecting the solution to the original ground set. Furthermore, given cardinal preferences and a threshold Δ, we show that the same algorithm can be used to find a 1.5-approximation for the maximum size Δ-min-stable common independent set. The running time of the algorithm is quadratic in $|S|$, and linear in case of partition matroids (which corresponds to the many-to-many stable matching problem with parallel edges allowed).

Finally, in Section 4, we show that the same general framework can be used to obtain efficient 1.5-approximation algorithms for two other natural near-stability notions: Δ-sum-stability and Δ-max-stability. These notions and their motivation are described in detail in Section 4.

1.3 Related work

The *stable marriage problem with ties and incomplete lists* was first studied by Iwama et al. [10], who showed the NP-hardness of Max-SMTI. Since then, various algorithms have been proposed to improve the approximation ratio [11, 12, 13], and the current best ratio is 1.5 by a polynomial-time algorithm of McDermid [16], where the same ratio is attained by linear-time algorithms of Paluch [17, 18] and Király [14, 15]. The 1.5-approximability extends to the many-to-one matching setting [15] and the student-project allocation problem with ties [5].

The stable marriage problem also has generalizations in which constraints are imposed on the possible sets of edges. Biró et al. [3] studied the *college admissions problem with common quotas*. Yokoi [22] considered a many-to-many matching model with ties and laminar constraints and presented a 1.5-approximation algorithm for the generalized Max-SMTI. Its approximation analysis depends on the base orderability of laminar matroids and cannot extend to the general matroid setting.

Cardinal preferences in the context of stable matchings have been studied for several reasons. Pini et al. [19] analyzed manipulations consisting of falsely reporting preference values. Among other stability notions, they introduced *α-stability*, which is equivalent to our definition of Δ-min-stability. Anshelevich et al. [1] considered approximate stability from the point of view of social welfare. They defined various utility models, and α-stability with respect to these models. They gave price-of-anarchy bounds that depend on the value of α. Chen et al. [4] defined *local d-near-stability* and *global d-near-stability* based on swaps in the preference orders.

2 Existence of perfect matching of exchange edges in matroids

In this secion, we present our key tool, a result on exchange properties of matroids. Let $M = (S, \mathcal{I})$ be a matroid, where S is the ground set and $\mathcal{I}$ is the family of independent sets. Recall that a base is a maximum size independent set, while a circuit is an inclusionwise minimal dependent set. The *fundamental circuit* of an element $x \in S \setminus B$ for a base B, denoted by $C_B(x)$, is the unique circuit in $B + x$. By a slight abuse of notation, we will also use $C_I(x)$ for an independent set I and an element $x \in S \setminus I$ to denote the unique circuit in $I + x$ if it exists. Any pair of circuits satisfies the following property.

Proposition 1 (Strong circuit axiom) *If C, C' are circuits, $x \in C \setminus C'$, and $y \in C \cap C'$, then there is a circuit $C^* \subseteq C \cup C'$ such that $x \in C^*$ and $y \notin C^*$.*

If we have a strict linear order $\succ$ given on S, then the triple $M = (S, \mathcal{I}, \succ)$ is called an *ordered matroid*. A nice property of ordered matroids is that for any weight vector $w \in \mathbb{R}^S$ which satisfies $w_x > w_y \Leftrightarrow x \succ y$, the unique maximum weight base is the same. We call this base A the *optimal base* of $(S, \mathcal{I}, \succ)$; it is characterized by the property that the worst element of $C_A(x)$ is x for any $x \in S \setminus A$. A similar statement about arbitrary circuits can be easily seen using the strong circuit axiom as follows.

Lemma 2 *Let A be the optimal base of an ordered matroid $(S, \mathcal{I}, \succ)$, and let C be a circuit. Then, the worst element of C is in $S \setminus A$.*

Proof: Suppose for contradiction that there are circuits with worst element being in A, and choose such a circuit C with $|C \setminus A|$ being the smallest possible. Clearly, $C \setminus A \neq \emptyset$, since A is a base. Let $x \in C \setminus A$, and let C' be the fundamental circuit of x for A. By the optimality of A, every element $y \in C' - x$ satisfies $y \succ x$.

Let z be the worst element of C. Then $z \prec x$, so $z \notin C'$. By the strong circuit axiom, there is a circuit $C'' \subseteq C \cup C'$ such that $z \in C''$ and $x \notin C''$. This is a contradiction, because C'' also satisfies the property, and $|C'' \setminus A| < |C \setminus A|$. □

We are now ready to prove the theorem that will be our main tool in proving the approximation bounds for our algorithms. As far as we know, this result on exchanges has not been observed previously in the literature. A *block matroid* is a matroid whose ground set can be partitioned into two bases.

Theorem 3 *Let $M = (S, \mathcal{I}, \succ)$ be an ordered block matroid of rank r, with the property that the complement of the optimal base A is also a base, denoted by B. Then, there is a perfect matching $a_i b_i$ $(i \in [r])$ between A and B such that $a_i \succ b_i$ and $b_i \in C_B(a_i)$ for every $i \in [r]$.*

Proof: Let $G = (A, B; E)$ be the bipartite graph formed by the pairs $a \in A$, $b \in B$ such that $a \succ b$ and $b \in C_B(a)$. Suppose for contradiction that there exists a set $X \subseteq A$ such that $|X| > |\Gamma_G(X)|$, where $\Gamma_G(X) = \{b \in B : \exists a \in X, ab \in E\}$. Let $Y \coloneqq \Gamma_G(X)$.

Notice that if C is a circuit such that $C \cap A \subseteq X$, then the worst element of C is in Y. Indeed, by Lemma 2, the worst element of C is in B, and this element b must be in the fundamental circuit $C_B(a)$ for some $a \in C \cap A \subseteq X$, because otherwise we could obtain a circuit in B by repeatedly removing elements in A using the strong circuit axiom, which would contradict the independence of B. Hence, $ab \in E$, and therefore $b \in Y$.

For each $x \in X$, let C_x be a circuit such that $C_x \cap A \subseteq X$, x is the worst element in $C_x \cap A$, and, subject to this, the worst element of $C_x \cap B$ is best possible (note that C_x always exists because $C_B(x)$ satisfies the first two properties). Let $y(x) \in Y$ denote the worst element of $C_x \cap B$. Since $|X| > |Y|$, there are elements $x \in X$ and $x' \in X$ such that $x \prec x'$ and $y(x) = y(x') =: y$. Notice that $x \notin C_{x'}$ because the worst element of $C_{x'} \cap A$ is x'. By applying the strong circuit axiom for C_x and $C_{x'}$, we can obtain a circuit C with the following properties:

- $C \subseteq C_x \cup C_{x'} - y$, so $C \cap A \subseteq X$, and the worst element of $C \cap B$ is better than y
- $x \in C$, so the worst element $C \cap A$ is x.

These properties contradict the choice of the circuit C_x, since C would have been a better choice. This contradiction implies that X cannot exist, so there is a perfect matching in G by Hall's theorem. □

3 Matroid kernel algorithm for weakly ordered matroids

In this section, we give a simple 1.5-approximation algorithm for Max-Kernel. A similar algorithm was presented by Yokoi [22] for a generalization of Max-SMTI that included laminar constraints. Her proof relied crucially on the property that the matroids induced by the laminar constraints are base orderable. In contrast, our proof works for arbitrary weakly ordered matroids.

To show the flexibility and usefulness of the algorithm, we prove the approximation ratio for the more general problem of finding a maximum size Δ-*min-stabe* common independent set, as defined in subsection 1.1. Note that if all p_i values are positive and Δ is small enough, then Δ-min-stability is equivalent to being an (M_1, M_2)-kernel.

3.1 Description of the algorithm

Let $M_1 = (S, \mathcal{I}_1)$ and $M_2 = (S, \mathcal{I}_2)$ be matroids on the same ground set S. We use $C_I^1(u)$ and $C_I^2(u)$ to denote fundamental circuits in M_1 and M_2, respectively. Let $\Delta > 0$ be a positive threshold, and let $p_i(v) \geq 0$ $(v \in S, i \in \{1, 2\})$ be the cardinal preferences for the two matroids. In the following, we describe the 1.5-approximation algorithm for finding the maximum size Δ-min-stable common independent set. Essentially, the algorithm creates a new instance by replacing each element by 3 parallel elements, and defines strict linear orders on the extended ground set based on the preferences on the original ground set. For the obtained ordered matroids M_1^* and M_2^*, an (M_1^*, M_2^*)-kernel A^* can be found in $\mathcal{O}(|S|^2)$ time by Fleiner's algorithm. The set A returned by the algorithm is the projection of A^* to the original ground set S.

To complete the description of the algorithm, we have to define the strict linear orders. Let the extended ground set be $S^* \coloneqq \cup_{u \in S} \{x_u, y_u, z_u\}$. We define the ordered matroid $M_i^* = (S^*, \mathcal{I}_i^*, \succ_i^*)$ as follows. The elements x_u, y_u, z_u are parallel in M_i^*, that is,

$$\mathcal{I}_i^* = \{\, I^* \subseteq S^* : \pi(I^*) \in \mathcal{I},\ |I^* \cap \{x_u, y_u, z_u\}| \leq 1\ \forall u \in S \,\},$$

where $\pi(I^*) = \{\, u \in S : I^* \cap \{x_u, y_u, z_u\} \neq \emptyset \,\}$. To define the linear orders $\succ_1^*$ and $\succ_2^*$, we first define cardinal preferences on the extended ground set as follows.

- $p_1^*(z_u) = p_1(u),\ \ p_1^*(y_u) = p_1(u) + K,\ \ p_1^*(x_u) = p_1(u) + K + \Delta$,
- $p_2^*(x_u) = p_2(u),\ \ p_2^*(y_u) = p_2(u) + K,\ \ p_2^*(z_u) = p_2(u) + K + \Delta$,

where K is a number larger than any $p_i(u)$ ($u \in S$, $i \in \{1, 2\}$). The linear orders $\succ_1^*$ and $\succ_2^*$ on S^* are obtained by considering the preference orders given by p_1^* and p_2^*, and breaking the ties so that $y_u \succ_1^* x_v$ if $p_1^*(y_u) = p_1^*(x_v)$ and $y_u \succ_2^* z_v$ if $p_2^*(y_u) = p_2^*(z_v)$ for any $u, v \in S$. This completes the construction of the ordered matroids M_1^* and M_2^*.

Lemma 4 *The output of our algorithm is a Δ-min-stable common independent set of M_1 and M_2.*

Proof: Let $A = \pi(A^*)$ be the output of the algorithm, where A^* is the (M_1^*, M_2^*)-kernel given by Fleiner's algorithm. It is clear from the definition that $A \in \mathcal{I}_1 \cap \mathcal{I}_2$. Suppose for contradiction that there exists $u \in S \setminus A$ that Δ-min-blocks A; we claim that y_u blocks A^*. Indeed, if $A + u \in \mathcal{I}_i$, then $A^* + y_u \in \mathcal{I}_i^*$, and if $p_i(u) \geq p_i(v) + \Delta$ for some $v \in C_A^i(u)$, then $v^* \coloneqq \{x_v, y_v, z_v\} \cap A^*$ satisfies $y_u \succ_i^* v^*$ and belongs to the fundamental circuit of y_u for A^* in M_i^*. □

3.2 Proof of 1.5-approximation

Theorem 5 *The approximation ratio of the above algorithm is at most* 1.5.

Proof: Let $A = \pi(A^*)$ be the output of the algorithm, where A^* is an (M_1^*, M_2^*)-kernel, and let B be a largest Δ-min-stable common independent set of M_1 and M_2. Suppose for contradiction that $|B| > 1.5|A|$. Let B_i be a subset of $B \setminus A$ such that $A \cup B_i \in \mathcal{I}_i$ and $|A \cup B_i| = |B|$ for each $i \in \{1, 2\}$. The sets B_1 and B_2 are disjoint because A^* is an inclusionwise maximal common independent set of M_1^* and M_2^*. In the following, we say that an element $u \in A$ is *of type x* (respectively y, z) if $\{x_u, y_u, z_u\} \cap A^* = x_u$ (respectively y_u, z_u).

Lemma 6 *Let $i \in \{1, 2\}$. There is a matching N_i of size $|B_{3-i}|$ between $A \setminus B$ and B_{3-i} such that the following hold for every $uv \in N_i$, where $u \in A$ and $v \in B$:*

1. *u is of type x or y if $i = 1$, and of type y or z if $i = 2$*
2. *$p_i(u) \geq p_i(v)$, and in particular $p_i(u) \geq p_i(v) + \Delta$ if u is of type y*
3. *either $v \in C_B^i(u)$ or $B + u \in \mathcal{I}_i$.*

Proof: Let $M' = (S', \mathcal{I}')$ be the matroid obtained from M_i by deleting $S \setminus (A \cup B)$, contracting $(A \cap B) \cup B_i$, and truncating to the size of $A \setminus B$. That is, $S' = (A \setminus B) \cup (B \setminus (A \cup B_i))$ and $\mathcal{I}' = \{\, I \subseteq S' : I \subseteq A \cup B,\ I \cup (A \cap B) \cup B_i \in \mathcal{I}_i,\ |I| \leq |A \setminus B| \,\}$. In M', the sets $A' \coloneqq A \setminus B$ and $B' \coloneqq B \setminus (A \cup B_i)$ are bases that are complements of each other. We define a strict preference order $\succ'$ on S' in the following way. The elements of $B \setminus (A \cup B_i \cup B_{3-i})$ are worst (in arbitrary order). On the remaining elements, i.e., on the elements of $(A \setminus B) \cup B_{3-i}$, we define the preferences based on the strict preferences $\succ_i^*$ on S^*. To do this, we assign an element $u^* \in S^*$ to each $u \in (A \setminus B) \cup B_{3-i}$ as follows. Let $u^* = \{x_u, y_u, z_u\} \cap A^*$ if $u \in A \setminus B$, let $u^* = x_u$ if $i = 1$ and $u \in B_2$, and let $u^* = z_u$ if $i = 2$ and $u \in B_1$. We then let $u \succ' v$ if and only if $u^* \succ_i^* v^*$. In the ordered matroid $M' = (S', \mathcal{I}', \succ')$, A' is an optimal base. Indeed, v is the worst element of $C_{A'}(v)$ for every $v \in B'$. It is clear for the elements in $B' \setminus B_{3-i}$ by the definition of $\succ'$. As for each $v \in B_{3-i}$, since $A^* + v^* \in \mathcal{I}_{3-i}^*$ holds and A^* is an (M_1^*, M_2^*)-kernel, v^* must be the worst element of its fundamental circuit for A^*. By Theorem 3, there is a perfect matching N' between A' and B' such that $u \succ' v$ and $v \in C'_{B'}(u)$ for every $uv \in N'$, where $u \in A'$ and $v \in B'$.

Let N_i be the subset of N' induced by $A \cup B_{3-i}$. Then $|N_i| = |B_{3-i}|$, and the first two properties of the lemma are satisfied for every $uv \in N_i$, because $u \succ' v$. We now show that for every $uv \in N_i$, either $v \in C^i_B(u)$ or $B + u \in \mathcal{I}_i$. Since $v \in C'_{B'}(u)$, v is in the fundamental circuit of u for B' in the matroid obtained by truncating M_i to the size of $A \setminus B$. This means that it is either in the fundamental circuit also in M_i, or $B + u$ is independent in M_i, as required. □

We are now ready to prove the theorem by getting a contradiction. Since $|B| > 1.5|A|$ implies $|N_i| = |B_{3-i}| > |A \setminus B|/2$ ($i \in \{1, 2\}$), there is an element $u \in A \setminus B$ that is covered by both N_1 and N_2. Let $uv_1 \in N_1$, $uv_2 \in N_2$. Since the first two properties hold for $i \in \{1, 2\}$, u must be of type y, and $p_i(u) \geq p_i(v_i) + \Delta$ for $i \in \{1, 2\}$. But this means that u is a Δ-min-blocking element for B because of the third property, a contradiction. □

4 Extensions to other variants of Δ-stability

As a motivation, let us consider the special case of partition matroids, which corresponds to a two-sided matching problem with cardinal preferences. The motivation for Δ-min-stability is that blocking may require some effort from the agents, so we say that a pair does not block unless they *both* achieve a value increase of Δ. However, there are other natural requirements that we can associate to a given positive threshold Δ. In some applications, blocking can be viewed as a combined effort of a pair, so we may require that the *sum* of their increase in value should be at least Δ. In other applications, an extra effort may be required by the agent who initiates the blocking, so we might say that a pair does not block unless *one of them* achieves a value increase of Δ.

In this section, we introduce the precise definitions for the above-mentioned variants, called Δ-*sum-stability* and Δ-*max-stability*. For each of these stability concepts, finding a largest solution is NP-hard as it is a generalization of Max-SMTI. We show that, by a suitable modification of the construction of the extended ground set and the ordered matroids M_1^* and M_2^*, we obtain 1.5-approximation algorithms for these variants of Δ-stability, too.

4.1 Δ-sum-stability

Recall that we are given two matroids $M_1 = (S, \mathcal{I}_1)$ and $M_2 = (S, \mathcal{I}_2)$, as well as preferences defined by nonnegative values $p_i(v)$ for $v \in S$, $i \in \{1, 2\}$. Let Δ be a positive threshold.

Definition 7 *Let X be a common independent set of M_1 and M_2. An element $u \in S \setminus X$ is Δ-sum-blocking for X if the following hold:*

- *either $X + u \in \mathcal{I}_1$ or there is an element $v_1 \in X$ such that $p_1(u) > p_1(v_1)$ and $v_1 \in C^1_X(u)$,*
- *either $X + u \in \mathcal{I}_2$ or there is an element $v_2 \in X$ such that $p_2(u) > p_2(v_2)$ and $v_2 \in C^2_X(u)$,*
- *$p_1(u) - p_1(v_1) + p_2(u) - p_2(v_2) \geq \Delta$, where we take $v_i = \emptyset$ if $X + u \in \mathcal{I}_i$.*

A common independent set X is Δ-sum-stable if there is no Δ-sum-blocking element for X.

Our aim is to give an efficient 1.5-approximation algorithm for the problem of finding the largest Δ-sum-stable common independent set. As in the case of Δ-min-stability, we create an instance of the matroid kernel problem by adding parallel elements, and by defining strict linear orders on the extended ground set based on the preferences. However, the number of parallel elements will depend on the possible differences in the preference values.

Let $0 < d_1 < d_2 < \cdots < d_k < \Delta$ be the set of numbers strictly between 0 and Δ that can be obtained in the form $p_1(u) - p_1(v)$, $p_2(u) - p_2(v)$, $\Delta - p_1(u) + p_1(v)$, or $\Delta - p_2(u) + p_2(v)$ for some $u, v \in S \cup \{\emptyset\}$. Furthermore, let $d_0 = 0$. Clearly, $k \leq \mathcal{O}(|S|^2)$. It is also easy to observe that $d_{k-\ell+1} = \Delta - d_\ell$ for $1 \leq \ell \leq k$ by the symmetry over Δ in the definition.

For each element $u \in S$, we make $k+2$ parallel copies of u. Their set is denoted by $X_u^* := \{x_0(u), x_1(u), \ldots, x_{k+1}(u)\}$. Let $S^* := \cup_{u \in S} X_u^*$, and let the resulting two matroids on S^* be M_1^* and M_2^*. The families of independent sets are given by

$$\mathcal{I}_i^* = \{\, I^* \subseteq S^* : \pi(I^*) \in \mathcal{I},\ |I^* \cap X_u^*| \leq 1\ \forall u \in S \,\},$$

where $\pi(I^*) = \{\, u \in S : I^* \cap X_u^* \neq \emptyset \,\}$.

Next, we define the linear orders $\succ_1^*$ and $\succ_2^*$. We first define cardinal preferences on the extended ground set as follows.

- $p_1^*(x_{k+1}(u)) = p_1(u),\ \ p_1^*(x_\ell(u)) = p_1(u) + K + \Delta - d_\ell$ for $0 \leq \ell \leq k$,
- $p_2^*(x_0(u)) = p_2(u),\ \ p_2^*(x_\ell(u)) = p_2(u) + K + \Delta - d_{k-\ell+1}$ for $1 \leq \ell \leq k+1$,

where K is a number larger than any $p_i(u)$ ($u \in S$, $i \in \{1,2\}$). The linear orders $\succ_1^*$ and $\succ_2^*$ on S^* are obtained by considering the preference orders given by p_1^* and p_2^*, and breaking the ties according to the following rule:

- If $p_1^*(x_j(u)) = p_1^*(x_\ell(v))$ and $j < \ell$, then $x_j(u) \prec_1^* x_\ell(v)$,
- If $p_2^*(x_j(u)) = p_2^*(x_\ell(v))$ and $j < \ell$, then $x_j(u) \succ_2^* x_\ell(v)$.

This completes the construction of the ordered matroids M_1^* and M_2^*.

Theorem 8 *Let A^* be an (M_1^*, M_2^*)-kernel, and let $A = \pi(A^*)$. Then A is a Δ-sum-stable common independent set of M_1 and M_2.*

Proof: It is clear from the definition that A is a common independent set. Suppose there is a Δ-sum-blocking element $u \in S$. If either $A + u \in \mathcal{I}_1$ or $A + u \in \mathcal{I}_2$, then $x_{k+1}(u)$ or $x_0(u)$ blocks A^* respectively, a contradiction. Otherwise let v_1, v_2 be as in Definition 7. Let $v_i^* = A^* \cap X_{v_i}^*$ $(i = 1, 2)$.

As u is Δ-sum blocking, we have $p_1(u) - p_1(v_1) > 0$, $p_2(u) - p_2(v_2) > 0$, and $p_1(u) - p_1(v_1) + p_2(u) - p_2(v_2) \geq \Delta$. If $p_1(u) - p_1(v_1) \geq \Delta$, then we claim that $x_k(u)$ blocks A^*. Indeed, on one hand, $p_1(u) - d_k > p_1(u) - \Delta \geq p_1(v_1) = p_1(v_1) - d_0$, so $x_k(u) \succ_1^* x_0(v_1) \succeq_1^* v_1^*$. On the other hand, $p_2(u) - p_2(v_2) > 0$ implies $p_2(u) - d_1 \geq p_2(v_2)$, so $x_k(u) \succ_2^* x_{k+1}(v_2) \succeq_2^* v_2^*$.

A similar argument shows that if $p_2(u) - p_2(v_2) \geq \Delta$, then $x_1(u)$ blocks A^*. Now consider the case when both are smaller than Δ. We have $p_1(u) - p_1(v_1) = d_j$, $p_2(u) - p_2(v_2) = d_\ell$ for some $0 < j, \ell \leq k$ such that $d_j + d_\ell \geq \Delta$.

We claim that $x_j(u)$ blocks A^*. First, $p_1(u) - d_j = p_1(v_1) = p_1(v_1) - d_0$, so $x_j(u) \succ_1^* x_0(v_1) \succeq_1^* v_1^*$. Second, $p_2(u) - d_{k-j+1} = p_2(u) + d_j - \Delta \geq p_2(u) - d_\ell = p_2(v_2) = p_2(v_2) - d_0$, so $x_j(u) \succ_2^* x_{k+1}(v_1) \succeq_2^* v_2^*$.

Since $A^* - v_i^* + x_j(u) \in \mathcal{I}_i^*$ for $i = 1, 2$, this means that $x_j(u)$ blocks A^*, a contradiction. □

Theorem 9 *Let A^* be an (M_1^*, M_2^*)-kernel, let $A = \pi(A^*)$, and let B be a maximum size Δ-sum-stable common independent set for M_1 and M_2. Then $|B| \leq 1.5|A|$.*

Proof: Suppose for contradiction that $|B| > 1.5|A|$. We use a similar construction as in the proof of Theorem 5. Let B_i be a subset of $B \setminus A$ such that $A \cup B_i \in \mathcal{I}_i$ and $|A \cup B_i| = |B|$ for $i \in \{1, 2\}$. The sets B_1 and B_2 are disjoint because A^* is an inclusionwise maximal common independent set of M_1^* and M_2^*.

The following lemma is analogous to Lemma 6. For $u \in A$, let $u^* := A^* \cap X_u^*$.

Lemma 10 *Let $i \in \{1, 2\}$. There is a matching N_i of size $|B_{3-i}|$ between $A \setminus B$ and B_{3-i} such that the following hold for every $uv \in N_i$, where $u \in A$ and $v \in B$:*

1. *$u^* = x_j(u)$ for some $j \leq k$ if $i = 1$, and for some $j \geq 1$ if $i = 2$,*
2. *$p_i(u) \geq p_i(v) + d_j$ if $i = 1$, and $p_i(u) \geq p_i(v) + d_{k-j+1}$ if $i = 2$,*

3. either $v \in C^i_B(u)$ or $B + u \in \mathcal{I}_i$.

PROOF: Let $M' = (S', \mathcal{I}')$ be the same matroid as in the proof of Lemma 6, and let $A' \coloneqq A \setminus B$, $B' \coloneqq B \setminus (A \cup B_i)$. We define a strict preference order $\succ'$ on S' in the following way. The elements of $B \setminus (A \cup B_i \cup B_{3-i})$ are worst (in arbitrary order). On the remaining elements, i.e., on the elements of $(A \setminus B) \cup B_{3-i}$, we define the preferences based on the strict preferences $\succ^*_i$ on S^*. Let $v^* = x_0(v)$ if $i = 1$ and $v \in B_2$, and let $v^* = x_{k+1}(v)$ if $i = 2$ and $v \in B_1$. Let $u \succ' v$ if and only if $u^* \succ^*_i v^*$. As in the proof of Lemma 6, A' is an optimal base in the ordered matroid $M' = (S', \mathcal{I}', \succ')$. By Theorem 3, there is a perfect matching N' between A' and B' such that $u \succ' v$ and $v \in C'_{B'}(u)$ for every $uv \in N'$, where $u \in A'$ and $v \in B'$.

Let N_i be the subset of N' induced by $A \cup B_{3-i}$. Consider $uv \in N_i$, $u \in A'$, $v \in B_{3-i}$. The first property in the lemma holds because $x_{k+1}(u) \prec^*_1 x_0(v)$ and $x_0(u) \prec^*_2 x_{k+1}(v)$ by the definitions of $\succ^*_1$ and $\succ^*_2$. To see the second property in the case $i = 1$, observe that $u \succ' v$ implies $x_j(u) \succ^*_1 x_0(v)$, thus $p_1(u) - d_j \geq p_1(v)$. Similarly, in the case $i = 2$, $u \succ' v$ implies $x_j(u) \succ^*_2 x_{k+1}(v)$, thus $p_1(u) - d_{k-j+1} \geq p_1(v)$.

The third property follows similarly to the proof of Lemma 6. □ Since $|B| > 1.5|A|$, there is an element $u \in A \setminus B$ that is covered by both N_1 and N_2. Let v_1, v_2 be u's partners in N_1 and N_2 respectively. By the first property of the lemma, $u^* = x_j(u)$ for some $j \in \{1, \ldots, k\}$. By the second property, $p_1(u) \geq p_1(v_1) + d_j$ and $p_2(u) \geq p_2(v_2) + d_{k-j+1}$.

Using that $d_j > 0$, $d_{k-j+1} > 0$, $d_j + d_{k-j+1} = \Delta$, and the fact that $v_i \in C^i_B(u)$ or $B + u \in \mathcal{I}_i$ for $i = 1, 2$, we get that the element u is Δ-sum-blocking for B, a contradiction. □

4.2 Δ-max-stability

The proofs in this section are very similar to those for Δ-sum-stability, so we skip some details that are identical. Let Δ be a positive threshold.

Definition 11 *Let X be a common independent set of M_1 and M_2. An element $u \in S \setminus X$ is Δ-max-blocking for X if the following hold:*

- *either $X + u \in \mathcal{I}_1$ or there is an element $v_1 \in X$ such that $p_1(u) > p_1(v_1)$ and $v_1 \in C^1_X(u)$,*
- *either $X + u \in \mathcal{I}_2$ or there is an element $v_2 \in X$ such that $p_2(u) > p_2(v_2)$ and $v_2 \in C^2_X(u)$,*
- *$\max\{p_1(u) - p_1(v_1), p_2(u) - p_2(v_2)\} \geq \Delta$, where we take $v_i = \emptyset$ if $X + u \in \mathcal{I}_i$.*

A common independent set X is Δ-max-stable if there is no Δ-max-blocking element for X.

In order to get a 1.5-approximation, we create a matroid kernel instance by taking four parallel elements of each element $u \in S$, denoted by $X^*_u \coloneqq \{x_0(u), x_1(u), x_2(u), x_3(u)\}$. Let $S^* = \cup_{u \in S} X^*_u$. As in the previous section, the families of independent sets are given by

$$\mathcal{I}^*_i = \{\, I^* \subseteq S^* : \pi(I^*) \in \mathcal{I},\ |I^* \cap X^*_u| \leq 1\ \forall u \in S \,\},$$

where $\pi(I^*) = \{\, u \in S : I^* \cap X^*_u \neq \emptyset \,\}$. We introduce the following cardinal preferences on S^*:

- $p^*_1(x_3(u)) = p_1(u)$, $p^*_1(x_2(u)) = p_1(u) + K$, $p^*_1(x_1(u)) = p^*_1(x_0(u)) = p_1(u) + K + \Delta$,
- $p^*_2(x_0(u)) = p_2(u)$, $p^*_2(x_1(u)) = p_2(u) + K$, $p^*_2(x_2(u)) = p^*_2(x_3(u)) = p_2(u) + K + \Delta$,

where K is larger than any $p_i(u)$ ($u \in S$, $i \in \{1, 2\}$). The linear orders $\succ^*_1$ and $\succ^*_2$ on S^* are obtained by considering the preference orders given by p^*_1 and p^*_2, and breaking the ties according to the following rule:

- If $p^*_1(x_j(u)) = p^*_1(x_\ell(v))$ and $j < \ell$, then $x_j(u) \prec^*_1 x_\ell(v)$, except for $x_0(u) \succ^*_1 x_1(v)$

- If $p_2^*(x_j(u)) = p_2^*(x_\ell(v))$ and $j < \ell$, then $x_j(u) \succ_2^* x_\ell(v)$, except for $x_2(u) \prec_2^* x_3(v)$

This completes the construction of the ordered matroids M_1^* and M_2^*.

Theorem 12 *Let A^* be an (M_1^*, M_2^*)-kernel, and let $A = \pi(A^*)$. Then A is a Δ-max-stable common independent set of M_1 and M_2.*

Proof: It is clear from the definition that A is a common independent set. Suppose there is a Δ-max-blocking element $u \in S$. If either $A + u \in \mathcal{I}_1$ or $A + u \in \mathcal{I}_2$, then $x_3(u)$ or $x_0(u)$ blocks A^* respectively. Otherwise let v_1, v_2 be as in Definition 11. Let $v_i^* = A^* \cap X_{v_i}^*$ $(i = 1, 2)$.

As u is Δ-max-blocking, we have $p_1(u) - p_1(v_1) > 0$, $p_2(u) - p_2(v_2) > 0$, and $\max\{p_1(u) - p_1(v_1), p_2(u) - p_2(v_2)\} \geq \Delta$. By symmetry, we may assume that $p_1(u) - p_1(v_1) \geq \Delta$.

We claim that $x_2(u)$ blocks A^*. First, $p_1(u) \geq p_1(v_1) + \Delta$, so $p_1^*(x_2(u)) \geq p_1^*(v_1^*)$, and equality may hold only when $v_1^* \in \{x_0(v_1), x_1(v_1)\}$. By the tie-breaking rule, we have $x_2(u) \succ_1^* v_1^*$. Second, $p_2(u) > p_2(v_2)$, so $p_2^*(x_2(u)) > p_2^*(v_2^*)$, which means that $x_2(u) \succ_2^* v_2^*$.

Since $A^* - v_i^* + x_2(u) \in \mathcal{I}_i^*$ for $i = 1, 2$, this means that $x_2(u)$ blocks A^*, a contradiction. □

Theorem 13 *Let A^* be an (M_1^*, M_2^*)-kernel, let $A = \pi(A^*)$, and let B be a maximum size Δ-max-stable common independent set for M_1 and M_2. Then $|B| \leq 1.5|A|$.*

Proof: Suppose for contradiction that $|B| > 1.5|A|$. Let B_i be a subset of $B \setminus A$ such that $A \cup B_i \in \mathcal{I}_i$ and $|A \cup B_i| = |B|$ for $i \in \{1, 2\}$. The sets B_1 and B_2 are disjoint because A^* is an inclusionwise maximal common independent set of M_1^* and M_2^*. For $u \in A$, we use the notation $u^* \coloneqq A^* \cap X_u^*$.

Lemma 14 *Let $i \in \{1, 2\}$. There is a matching N_i of size $|B_{3-i}|$ between $A \setminus B$ and B_{3-i} such that the following hold for every $uv \in N_i$, where $u \in A$ and $v \in B$:*

1. *$u^* = x_j(u)$ for some $j \leq 2$ if $i = 1$, and for some $j \geq 1$ if $i = 2$,*
2. *$p_i(u) > p_i(v)$ if either $i = j = 1$ or $i = j = 2$,*
3. *$p_i(u) \geq p_i(v) + \Delta$ if either $i = 1$ and $j = 2$, or $i = 2$ and $j = 1$,*
4. *either $v \in C_B^i(u)$ or $B + u \in \mathcal{I}_i$.*

Proof: Let $M' = (S', \mathcal{I}')$ be the same matroid as in the proof of Lemma 6, and let $A' \coloneqq A \setminus B$, $B' \coloneqq B \setminus (A \cup B_i)$. We define a strict preference order $\succ'$ on S' in the following way. The elements of $B \setminus (A \cup B_i \cup B_{3-i})$ are worst (in arbitrary order). On the remaining elements, i.e., on the elements of $(A \setminus B) \cup B_{3-i}$, we define the preferences based on the strict preferences $\succ_i^*$ on S^*. Let $v^* = x_0(v)$ if $i = 1$ and $v \in B_2$, and let $v^* = x_3(v)$ if $i = 2$ and $v \in B_1$. Let $u \succ' v$ if and only if $u^* \succ_i^* v^*$. With these preferences, A' is an optimal base in the ordered matroid $M' = (S', \mathcal{I}', \succ')$. By Theorem 3, there is a perfect matching N' between A' and B' such that $u \succ' v$ and $v \in C'_{B'}(u)$ for every $uv \in N'$, where $u \in A'$ and $v \in B'$.

Let N_i be the subset of N' induced by $A \cup B_{3-i}$. Consider $uv \in N_i$, $u \in A'$, $v \in B_{3-i}$. The first property in the lemma holds because $x_3(u) \prec_1^* x_0(v)$ and $x_0(u) \prec_2^* x_3(v)$. To see the second and third properties in the case $i = 1$, observe that $u \succ' v$ implies $x_j(u) \succ_1^* x_0(v)$. By the tiebreaking rule, this implies $p_1(u) > p_1(v)$ if $j = 1$, and $p_1(u) \geq p_1(v) + \Delta$ if $j = 2$. Similarly, in the case $i = 2$, we get $p_2(u) > p_2(v)$ if $j = 2$, and $p_2(u) \geq p_2(v) + \Delta$ if $j = 1$. The fourth property follows similarly to the proof of Lemma 6. □ Since $|B| > 1.5|A|$, there is an element $u \in A \setminus B$ that is covered by both N_1 and N_2. Let v_1, v_2 be u's partners in N_1 and N_2 respectively. By the first property of the lemma, $u^* = x_j(u)$ for some $j \in \{1, 2\}$. If $j = 1$, then the second and third properties imply that $p_1(u) > p_1(v_1)$ and $p_2(u) \geq p_2(v_2) + \Delta$. If $j = 2$, then we get $p_1(u) \geq p_1(v_1) + \Delta$ and $p_2(u) > p_2(v_2)$.

By the fourth property of Lemma 14, we obtain that the element u is Δ-sum-blocking for B, a contradiction. □

References

[1] Elliot Anshelevich, Sanmay Das, and Yonatan Naamad. Anarchy, stability, and utopia: creating better matchings. *Autonomous Agents and Multi-Agent Systems*, 26(1):120–140, 2013.

[2] Haris Aziz, Péter Biró, and Makoto Yokoo. Matching market design with constraints. In *Proc. of 36th AAAI Conference on Artificial Intelligence (AAAI 2022)*, volume 36, pages 12308–12316, 2022.

[3] Péter Biró, Tamás Fleiner, Robert W Irving, and David F Manlove. The college admissions problem with lower and common quotas. *Theoretical Computer Science*, 411(34):3136–3153, 2010.

[4] Jiehua Chen, Piotr Skowron, and Manuel Sorge. Matchings under preferences: Strength of stability and tradeoffs. *ACM Transactions on Economics and Computation*, 9(4):1–55, 2021.

[5] Frances Cooper and David Manlove. A 3/2-approximation algorithm for the student-project allocation problem. In *Proc. 17th International Symposium on Experimental Algorithms (SEA 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

[6] Tamás Fleiner. A matroid generalization of the stable matching polytope. In *Proc. 8th International Conference on Integer Programming and Combinatorial Optimization*, pages 105–114. Springer, 2001.

[7] Tamás Fleiner. A fixed-point approach to stable matchings and some applications. *Mathematics of Operations research*, 28(1):103–126, 2003.

[8] David Gale and Lloyd S Shapley. College admissions and the stability of marriage. *American Mathematical Monthly*, 69(1):9–15, 1962.

[9] Chien-Chung Huang. Classified stable matching. In *Proc. twenty-first annual ACM-SIAM symposium on Discrete Algorithms (SODA 2010)*, pages 1235–1253. SIAM, 2010.

[10] Kazuo Iwama, David Manlove, Shuichi Miyazaki, and Yasufumi Morita. Stable marriage with incomplete lists and ties. In *Proc. 26th International Colloquium on Automata, Languages, and Programming (ICALP 1999)*, pages 443–452. Springer, 1999.

[11] Kazuo Iwama, Shuichi Miyazaki, and Naoya Yamauchi. A 1.875-approximation algorithm for the stable marriage problem. In *Proc. Eighteenth annual ACM-SIAM symposium on Discrete algorithms (SODA 2007)*, pages 288–297. SIAM, Philadelphia, 2007.

[12] Kazuo Iwama, Shuichi Miyazaki, and Naoya Yamauchi. A $(2 - c\frac{1}{\sqrt{n}})$-approximation algorithm for the stable marriage problem. *Algorithmica*, 51(3):342–356, 2008.

[13] Zoltán Király. Better and simpler approximation algorithms for the stable marriage problem. *Algorithmica*, 60(1):3–20, 2011.

[14] Zoltán Király. Linear time local approximation algorithm for maximum stable marriage. In *Proc. Second International Workshop on Matching Under Preferences (MATCH-UP 2012)*, page 99, 2012.

[15] Zoltán Király. Linear time local approximation algorithm for maximum stable marriage. *Algorithms*, 6(3):471–484, 2013.

[16] Eric McDermid. A 3/2-approximation algorithm for general stable marriage. In *Proc. 36th International Colloquium on Automata, Languages, and Programming (ICALP 2009)*, pages 689–700. Springer, 2009.

[17] Katarzyna Paluch. Faster and simpler approximation of stable matchings. In *Proc. 9th International Workshop on Approximation and Online Algorithms (WAOA 2011)*, pages 176–187, 2011.

[18] Katarzyna Paluch. Faster and simpler approximation of stable matchings. *Algorithms*, 7(2):189–202, 2014.

[19] Maria Silvia Pini, Francesca Rossi, K Brent Venable, and Toby Walsh. Stability, optimality and manipulation in matching problems with weighted preferences. *Algorithms*, 6(4):782–804, 2013.

[20] Alexander Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*, volume 24. Springer, 2003.

[21] Hiroki Yanagisawa. Approximation algorithms for stable marriage problems. *PhD thesis, Kyoto University, Graduate School of Informatics*, 2007.

[22] Yu Yokoi. An approximation algorithm for maximum stable matching with ties and constraints. In *Proc. 32nd International Symposium on Algorithms and Computation (ISAAC 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.

Fairness versus transparency in the UEFA Champions League: how to choose a random perfect matching in a balanced bipartite graph

László Csató

Research Group of Operations Research and Decision Systems
Laboratory on Engineering and Management Intelligence
Institute for Computer Science and Control (SZTAKI)

Department of Operations Research and Actuarial Sciences
Institute of Operations and Decision Sciences
Corvinus University of Budapest (BCE)

Budapest, Hungary

laszlo.csato@sztaki.hu

Abstract: The organiser of the UEFA Champions League, one of the most prestigious football tournaments around the world, faces a non-trivial mathematical and statistical problem each autumn: how to choose a random perfect matching in a balanced bipartite graph. Furthermore, since the set of edges depends on the national associations of the teams qualified for the Round of 16, the graph is almost guaranteed to change in every season. For the sake of credibility and transparency, the draw consists of some discrete uniform choices from two urns whose compositions are dynamically updated with computer assistance. Even though the adopted mechanism is unevenly distributed over all valid assignments, it resembles the fairest possible lottery according to a recent result. We challenge this finding by analysing the effect of reversing the order of the urns and show that the modification would have reduced the level of unfairness by 15-20% in certain seasons and the improvement can be close to 30%.

Keywords: constrained assignment; draw mechanism; fairness; tournament design; UEFA Champions League

1 Introduction

Finding a random perfect matching in a balanced bipartite graph, where the two subsets of nodes have the same cardinality, seems to be a purely mathematical problem. However, the Union of European Football Associations (UEFA) faces this challenge in the UEFA Champions League Round of 16 draw each autumn. In particular, the eight group winners and the eight runners-up that qualified for the knockout stage need to be divided into eight mutually disjoint pairs subject to the following restrictions:

- *Bipartite constraint*: a group winner is not allowed to play against a runner-up;
- *Group constraint*: teams from the same group cannot be paired;
- *Association constraint*: teams from the same country cannot be matched.

The first restriction, the bipartite constraint, reduces the number of valid assignments to $8! = 40320$ since every result of the draw can be represented by a permutation of the eight groups. The second restriction, the group constraint, means that only a derangement (a permutation without a fixed point, in other words, a permutation where no element appears in its original position) is allowed, which decreases

the number of valid assignments to 14833 [17]. Finally, the impact of the association constraint depends on the identity of the teams, thus, no simple combinatorial formula exists to determine the number of possible results of the draw. For instance, there have been 4781 feasible assignments in the 2020/21 season [13] but only 3876 in the 2022/2023 season [14].

In order to ensure ex-ante *fairness*, all valid assignments should occur with the same probability. It can be easily achieved by listing all feasible outcomes and drawing one of them randomly. However, UEFA does not use this procedure because of at least two reasons. First, it would be boring to watch, which is important as the draw ceremony is streamed live over the internet and broadcast by several national media companies [1]. Second, it would be impossible to detect fraud and prevent conspiracy theories: checking credibility is a crucial aspect since the ex-post result will certainly favour some teams at the expense of others.

Therefore, UEFA has adopted an easy to follow randomisation mechanism: the names of the teams are physically placed in two urns and the composition of the urns is dynamically updated with computer assistance to ensure that all constraints will be satisfied. The draw consists of discrete uniform choices from the urns that can be observed. Even though the computer-assisted algorithm is essentially a black box and carries out non-trivial calculations, all computations are deterministic and can be verified during or after the draw with basic mathematical knowledge. For instance, a mistake was detected in the draw of the 2021/22 season, and the whole process was repeated three hours later [12].

Naturally, there is no "free lunch" and the randomisation procedure chosen by UEFA is unfair, i.e., the feasible outcomes are not equally likely [16, 17]. However, utilising the main result of [2], [1] verify that the design of the UEFA Champions League Round of 16 draw is near-optimal: it is close to a constrained-best solution as it resembles the fairest possible lottery over the constrained assignments.

We challenge or, at least, refine this important finding of [1] by analysing the effect of reversing the order of the urns. It is shown that a straightforward modification of the draw procedure would have reduced the level of unfairness by 15-20% in certain seasons and has the potential to improve by almost 30%.

Our results are unexpected to some degree because, even though the impact of the draw order has been recognised [1, Proposition 2], it has been called marginal in [17, Footnote 19] and has been commented as slight for the 2017/18 [9], 2019/20 [11], and 2022/23 seasons [14]. Consequently, the current study offers a novel example of how operations research and applied mathematics can contribute to the design of sports tournaments [22, 3].

The paper is organised as follows. Section 2 describes the rules of the UEFA Champions League Round of 16 draw. The research problem and the methodology are introduced in Section 3, while the main contributions are presented in Section 4. Section 5 summarises our findings and raises open questions.

2 The UEFA Champions League Round of 16 draw

The UEFA Champions League is one of the most prestigious football tournaments around the world. It is contested by the leading European clubs that can qualify primarily based on the results of their domestic leagues in the previous season. Since the 1997/98 season, multiple entrants are allowed from certain countries; now the strongest leagues can provide up to five teams as has happened for Germany in 2022/23.

The Champions League is organised in the same format since the 2003/04 season: a group stage played in eight home-away round-robin groups, from which the top two teams qualify for the knockout stage starting from the Round of 16. The knockout phase consists of two-legged clashes, each team plays one game home and one away, except for the final, which is played in a predetermined neutral field. In the Round of 16 draw, three restrictions (the bipartite, the group, and the association constraint) apply as detailed in Section 1. On 17 July 2014, the UEFA emergency panel decided that Ukrainian and Russian clubs could not be drawn against each other "until further notice" due to political reasons. This rule has been effective only in the 2015/16 season, and can be treated similar to the association constraint by

Table 1: Teams playing in the 2012/13 UEFA Champions League Round of 16

Group	Runner-up		Group winner	
	Club	Country	Club	Country
A	Porto	Portugal	Paris Saint-Germain	France
B	Arsenal	England	Schalke 04	Germany
C	Milan	Italy	Málaga	Spain
D	Real Madrid	Spain	Borussia Dortmund	Germany
E	Shakhtar Donetsk	Ukraine	Juventus	Italy
F	Valencia	Spain	Bayern München	Germany
G	Celtic Glasgow	Scotland	Barcelona	Spain
H	Galatasaray	Turkey	Manchester United	England

considering Russia and Ukraine as the same country. There are no draw restrictions in later rounds of the knockout stage, each team can meet all the others.

The Round of 16 draw is equivalent to selecting a perfect matching in a balanced bipartite graph with 2×8 nodes. UEFA uses the following mechanism for this purpose:

- Eight balls containing the names of the eight runners-up are placed in a bowl;
- A ball is drawn from the bowl, and the team drawn plays at home in match 1;
- The computer shows which group winners are eligible to play as the visiting team in match 1;
- Balls representing these teams are placed in another bowl;
- A ball is drawn from the second bowl to complete the pairing for match 1;
- The above procedure is repeated for the remaining matches.

The computer may indicate that only one group winner is allowed to play as the visiting team, in which case there is no need to draw the sole ball.

The mechanism is more complicated than it seems at first glance because the computer should check not only whether draw conditions apply for the runner-up chosen, but also whether draw conditions are anticipated to apply for the runner(s)-up still to be drawn. Let us see the following illustration.

Example 1 *[17] The teams qualifying for the 2012/13 UEFA Champions League Round of 16 are shown in Table 1. The draw happened as follows:*

1. *The runner-up Galatasaray was drawn first. Its eligible opponents were all teams except for Manchester United due to the group constraint. Out of the seven group winners, Schalke 04 was drawn.*
2. *The runner-up Celtic Glasgow was drawn second. Its possible opponents were all teams except for Schalke 04 (already drawn), and Barcelona (group constraint). Out of the six group winners, Juventus was drawn.*
3. *The runner-up Arsenal was drawn third. Its admissible opponents were all teams except for Schalke 04, Juventus (already drawn), and Manchester United (association constraint). Note that the group constraint was ineffective. Out of the five group winners, Bayern München was drawn.*
4. *The runner-up Shakhtar Donetsk was drawn fourth. Its eligible opponents were all teams except for Schalke 04, Juventus, and Bayern München (already drawn). Out of the five remaining group winners, Borussia Dortmund was drawn.*
5. *The runner-up Milan was drawn fifth. The computer indicated that it should be paired with Barcelona, thus, no draw was carried out.*

Does this indicate a flaw? Naturally not. At this point, four group winners remained to be drawn: Paris Saint-Germain, Málaga, Barcelona, Manchester United. Málaga was prohibited by the group constraint. If Milan would have played against the French or English team, then three pairings would have been left with four Spanish clubs (Real Madrid, Valencia, Málaga, Barcelona), and the association constraint certainly would have been violated.

However, two (Real Madrid, Valencia) out of the three remaining runners-up could have faced either Paris Saint-Germain or Manchester United, therefore, the draw was still not finished and remained interesting for the spectators.

In the following, the procedure above will be called the *standard UEFA mechanism*.

The randomisation procedure of the UEFA always leads to a valid matching if there is an assignment satisfying all constraints. The existence can be proved by Hall's marriage theorem analogously to [16], but now the maximal number of teams in the Round of 16 is five for at most two countries. [15, Chapter 3.6] mentions this result as an application of mathematics in everyday life.

The same algorithm is used in other UEFA club tournaments such as the UEFA Europa League and the UEFA Europa Conference League [5]. Furthermore, it is applied to divide teams into groups with more than two teams in several competitions for national teams: the FIBA Basketball World Cup [6], the UEFA Nations League [20], the UEFA Euro qualifying [21], and the European Qualifiers to the FIFA World Cup [19]. Last but not least, after a reform inspired by the criticism of [8], the FIFA World Cup draw is also carried out by this randomisation procedure since 2018 [10, 4].

3 Research question and methodology

The UEFA mechanism is known to be not evenly distributed, it is distinct from a uniform draw over all admissible matchings [1, 7, 16, 17]. Even though the distortions seem to be small with respect to the differences of pairwise match probabilities, it is quite frequent that a certain club i plays against club j with a higher probability than against club k according to a uniform draw, but a match between i and k has a higher chance to occur than a match between i and j according to the UEFA mechanism [17]. In addition, even the small probability differences may change the expected revenue of some teams by more than 10 thousand euros due to the substantial amount of prize money [17].

Hence, the outcome needs to be as close to the fair uniform draw as possible. However, all oddities discussed above would still be present if at every point during the procedure, a random choice would be made whether to match a winner to a runner-up or vice versa [17, Footnote 19]. Furthermore, [1] examine all possible counterfactual lotteries over the feasible assignments to conclude that the design of the draw mechanism is near-optimal. Nonetheless, it is worth analysing whether the *reversed UEFA mechanism*, where the group winners are drawn first instead of the runners-up, is able to yield a fairer outcome.

Let us see the implied probabilities of a match between two clubs in the 2012/13 UEFA Champions League Round of 16 draw.

Example 2 *Consider the three draw procedures (uniform choice among all valid assignments, standard UEFA mechanism, reversed UEFA mechanism) in Example 1 with the teams listed in Table 1.*

Table 2 presents the ideal probabilities according to a perfect draw. The association constraint is effective for six pairs: Arsenal and Manchester United (winner of Group H), Milan and Juventus (winner of Group E), Real Madrid and Málaga (winner of Group C), Real Madrid and Barcelona, Valencia and Málaga, Valencia and Barcelona. There are 5463 valid matchings; among them, Porto plays against Schalke 04 (winner of Group B) in 636 cases, against Málaga (winner of Group C) in 1036 cases, and so on. Note that the probability of Porto vs Borussia Dortmund (winner of Group D) and Porto vs Bayern München (winner of Group F) coincide ($676/5463 \approx 12.37\%$*) because these two German group winners are symmetric as the runners-up in their groups are Spanish teams. On the other hand, the probabilities for Schalke 04 are different since the runner-up in its group is the English club Arsenal. Analogously, the matches Porto vs Manchester United and Arsenal vs Paris Saint-Germain (winner of Group A) have the same probability of* $731/5463 \approx 13.05\%$ *as Porto and Paris-Saint Germain both come from Group*

Table 2: Ideal probabilities for each pairing in the 2012/13 season

Runner-up	Group of the group winner							
	A	B	C	D	E	F	G	H
Porto	0	11.64%	18.96%	12.37%	13.34%	12.37%	18.25%	13.05%
Arsenal	13.05%	0	22.22%	14.17%	15.14%	14.17%	21.25%	0
Milan	14.48%	14.68%	0	15.54%	0	15.54%	23.19%	16.57%
Real Madrid	18.40%	18.78%	0	0	21.76%	19.55%	0	21.51%
Shakhtar Donetsk	11.68%	11.83%	19.31%	12.61%	0	12.61%	18.71%	13.25%
Valencia	18.40%	18.78%	0	19.55%	21.76%	0	0	21.51%
Celtic Glasgow	12.36%	12.52%	20.14%	13.20%	14.48%	13.20%	0	14.11%
Galatasaray	11.64%	11.77%	19.37%	12.56%	13.51%	12.56%	18.60%	0

Table 3: Probabilities for each pairing in the 2012/13 season, UEFA mechanism

Runner-up	Group of the group winner							
	A	B	C	D	E	F	G	H
Porto	0	11.68%	18.86%	12.21%	13.44%	12.21%	18.30%	13.30%
Arsenal	13.39%	0	21.69%	14.19%	15.54%	14.20%	20.98%	0
Milan	14.37%	14.61%	0	15.48%	0	15.48%	23.47%	16.59%
Real Madrid	18.33%	18.71%	0	0	21.58%	20.16%	0	21.23%
Shakhtar Donetsk	11.70%	11.90%	19.38%	12.43%	0	12.40%	18.63%	13.55%
Valencia	18.33%	18.69%	0	20.12%	21.59%	0	0	21.26%
Celtic Glasgow	12.18%	12.38%	20.87%	13.15%	14.21%	13.14%	0	14.07%
Galatasaray	11.69%	12.03%	19.19%	12.42%	13.64%	12.41%	18.62%	0

Table 4: Probabilities for each pairing in the 2012/13 season, reversed UEFA mechanism

Runner-up	Group of the group winner							
	A	B	C	D	E	F	G	H
Porto	0	11.70%	18.85%	12.16%	13.49%	12.16%	18.27%	13.37%
Arsenal	13.29%	0	21.92%	14.10%	15.52%	14.09%	21.08%	0
Milan	14.37%	14.59%	0	15.48%	0	15.48%	23.47%	16.61%
Real Madrid	18.37%	18.69%	0	0	21.47%	20.38%	0	21.09%
Shakhtar Donetsk	11.72%	11.91%	19.37%	12.40%	0	12.38%	18.61%	13.62%
Valencia	18.35%	18.70%	0	20.36%	21.48%	0	0	21.11%
Celtic Glasgow	12.20%	12.39%	20.64%	13.13%	14.31%	13.12%	0	14.20%
Galatasaray	11.71%	12.02%	19.23%	12.37%	13.72%	12.39%	18.57%	0

A without an association constraint, while Arsenal and Manchester United are both English clubs such that the other team from their groups (Schalke 04 and Galatasaray, respectively) is without an association constraint and there are no more English teams. These numbers have already appeared in [16, Tabelle 2] and [17, Table 4].

Table 3 shows the probabilities according to the standard UEFA mechanism, approximated by 10 million simulated draws. This technique has been used by [1], too, as can be seen from [1, Table 1]. The greatest bias occurs for the pair Celtic Glasgow vs Málaga, which exceeds 0.73 percentage points. These numbers have already appeared in [16, Tabelle 3] and [17, Table 5] (the small differences are due to the inaccuracy of our simulations).

Finally, Table 4 contains the (simulated) probabilities according to the reversed UEFA mechanism. The greatest bias occurs for the pair Real Madrid vs Bayern München (winner of Group F), which exceeds

0.8 percentage points. These numbers have not appeared before in the literature, although the reversed randomisation procedure has already been presented for the 2017/18 [9], 2019/20 [11], and 2022/23 seasons [14], based on one million simulation runs.

The fairness of the standard and reversed UEFA mechanisms will be evaluated by taking the average of their biases for all pairs. Let p_{ij} be the ideal probability that clubs i and j are matched under the evenly distributed uniform draw, and p_{ij}^M be the probability of this event if mechanism M is used to obtain a feasible assignment. We define two fairness measures for mechanism M as follows:

$$F_1(M) = 1000 \times \frac{\sum_{i,j} \left| p_{ij} - p_{ij}^M \right|}{\sum_{i,j} \#\{p_{ij} > 0\}}, \tag{1}$$

$$F_2(M) = 10000 \times \frac{\sum_{i,j} \left(p_{ij} - p_{ij}^M \right)^2}{\sum_{i,j} \#\{p_{ij} > 0\}}, \tag{2}$$

where $\sum_{i,j} \#\{p_{ij} > 0\}$ is the number of team pairs with a positive probability. Its maximum is 56 if the association constraint is not effective, but this has never happened in the UEFA Champions League since 2003: the denominator varies between 43 (2019/20) and 53 (in five seasons). The multipliers of thousand and ten thousand in formulas (1) and (2), respectively, are normalising factors.

F_1 is called the *average absolute distortion* and F_2 is called the *average squared distortion*. Obviously, the latter is more strongly influenced by greater differences for certain team pairs.

[1] use a more complicated quantification of fairness, which measures the average absolute difference in the match likelihoods across all valid pairwise comparisons. However, their distortion metric equals zero only if there are no constraints in the draw, and increases with the number of restricted team pairs. On the other hand, both $F_1(M)$ and $F_2(M)$ are zero for the ideal evenly distributed uniform draw among all feasible matchings.

4 Results

Figure 1 reveals how our measures of fairness distortion depend on the number of restricted pairs due to the association constraint, which was six in Example 2. The bias of the UEFA procedure is somewhat greater if there are more effective constraints, but the relationship is rather weak. The standard and the reversed versions are not fundamentally different with respect to fairness, which is expected because they differ only in a small detail that is not expected to have a substantial effect as mentioned in [17, Footnote 19].

Nonetheless, the difference between the fairness of the standard and reversed UEFA mechanisms is non-negligible in some seasons according to Figure 2. For example, UEFA has had luck with its choice in the 2017/18 season, although the distortion has been among the highest this year. Contrarily, the reversed procedure would have been better in the next season of 2018/19. Overall, the unfairness of the two versions seems to be increasing in recent years.

That is partially caused by the higher number of exclusions as can be seen in Table 5. The table also uncovers that the standard (reversed) procedure has been the better option in 12 (8) seasons. The choice essentially does not depend on the measure of fairness, the bias of the two versions almost coincides in the two seasons (2016/17, 2020/21) when the order of metrics F_1 and F_2 differs. Selecting the fairer mechanism instead of the other would have reduced the distortion by more than 8% in the last 20 years.

Finally, note that the standard or the reversed UEFA randomisation procedure would be more favourable than its pair with the same chance from a purely mathematical point of view: the reversed version will be better if the sets of group-winners and runners-up are exchanged. Consequently, the reversed UEFA mechanism has the potential to improve fairness by more than 28% based on the 2017/18 season. However, this nice symmetry does not hold in the real-world if the clubs from some national associations are more likely to be group winners than runners-up or vice versa.

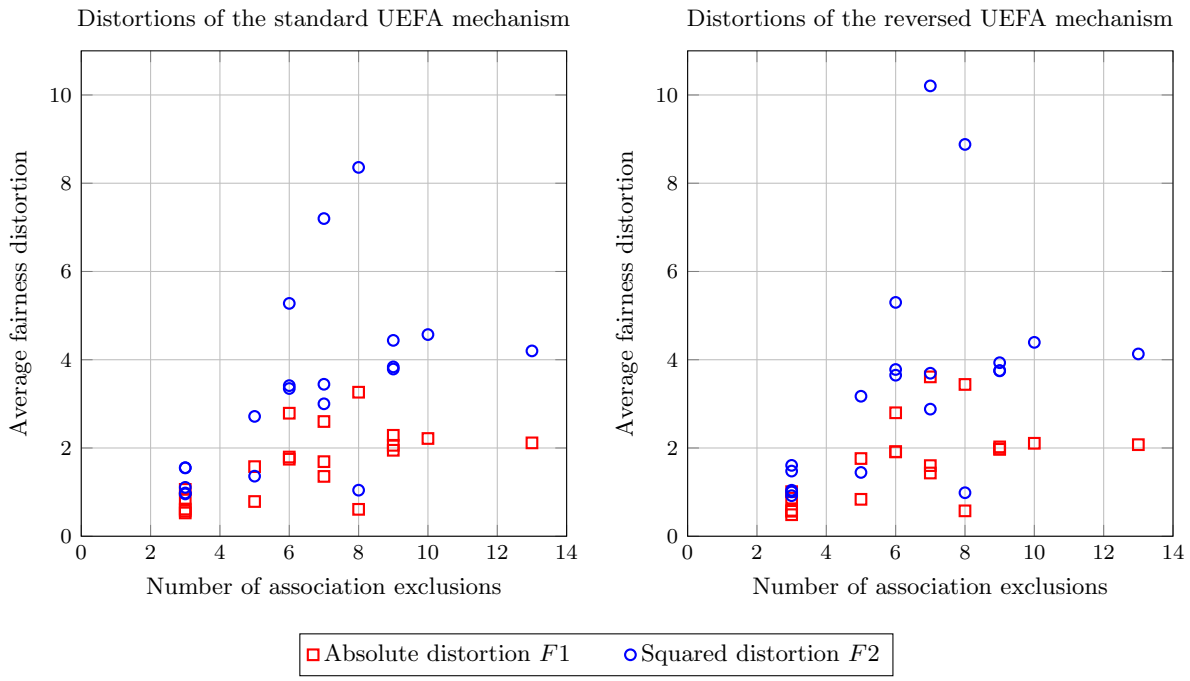


Figure 1:
The fairness of the standard and reversed UEFA mechanisms by the number of associaion exclusions

5 Conclusions

The paper has analysed a randomisation procedure used in a sports competition with high monetary stakes and substantial public interest. In particular, the UEFA Champions League Round of 16 draw is essentially equivalent to finding a random perfect matching in a balanced bipartite graph. In order to ensure credibility and transparency, the draw is implemented by a specific mechanism instead of choosing randomly an assignment that satisfies all criteria. Although the unfairness of the draw mechanism has been uncovered years ago [16, 17], the randomisation procedure adopted by the UEFA is near-optimal according to a recent paper published in a leading journal of management science [1]. This result has been refined by showing how reversing the arbitrarily chosen draw order can significantly reduce the level of unfairness.

There remains much scope for future research. First, the reasonfor the difference between the standard and the reversed UEFA mechanisms is unknown. Second, in contrast to [1], we think tournament organisers should continue the search for a fairer randomisation since the previous proposals [7, 17, 18] have different weaknesses.

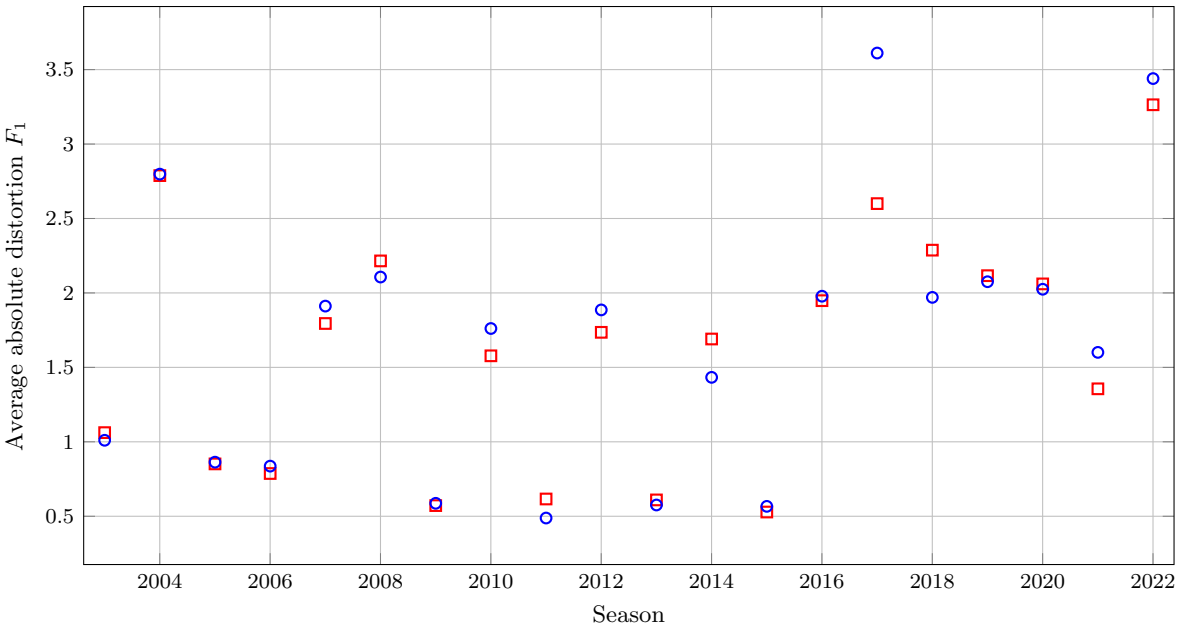
References

[1] M. Boczoń and A. J. Wilson, Goals, constraints, and transparently fair assignments: A field study of randomization design in the UEFA Champions League, *Management Science*, in press, DOI: 10.1287/mnsc.2022.4528 (2022)

[2] E. Budish, Y.-K. Che, F. Kojima, and P. Milgrom, Designing random allocation mechanisms: Theory and applications, *American Economic Review* **103**(2):585–623 (2013)

[3] L. Csató, *Tournament Design: How Operations Research Can Improve Sports Rules*, Palgrave Pivots in Sports Economics, Palgrave Macmillan, Cham, Switzerland (2021)

a Average absolute distortion F_1 of the randomisation mechanisms



b Average squared distortion F_2 of the randomisation mechanisms

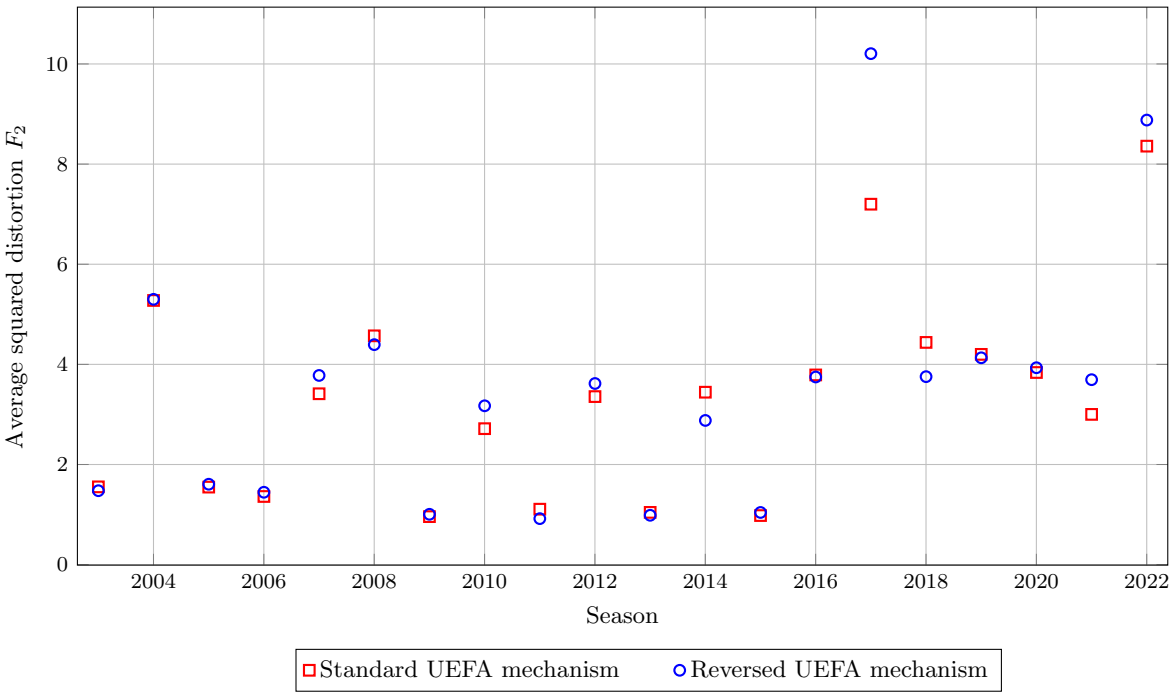


Figure 2: The comparison of fairness for the standard and reversed UEFA mechanisms by seasons
Note: The UEFA Champions League seasons are denoted by their first year since the Round of 16 draw is held in autumn.

Table 5: The relative performance of the standard and reversed UEFA mechanisms

Season	Number of valid assignments	Number of exclusions	Fairness measure F_1		Fairness measure F_2	
			Preferred	Reduction	Preferred	Reduction
2003/04	8476	3	Reversed	4.84%	Reversed	4.95%
2004/05	5427	6	Standard	0.36%	Standard	0.42%
2005/06	9200	3	Standard	1.35%	Standard	3.58%
2006/07	6655	5	Standard	5.92%	Standard	5.83%
2007/08	5271	6	Standard	6.11%	Standard	9.67%
2008/09	2988	10	Reversed	4.89%	Reversed	3.85%
2009/10	9094	3	Standard	2.62%	Standard	4.64%
2010/11	6304	5	Standard	10.41%	Standard	14.40%
2011/12	9147	3	Reversed	20.77%	Reversed	17.01%
2012/13	5463	6	Standard	8.92%	Standard	8.26%
2013/14	3497	8	Reversed	5.60%	Reversed	5.57%
2014/15	4516	7	Reversed	15.23%	Reversed	16.37%
2015/16	9147	3	Standard	6.73%	Standard	5.93%
2016/17	3501	9	Standard	1.52%	Reversed	1.08%
2017/18	4238	7	Standard	28.01%	Standard	29.46%
2018/19	3694	9	Reversed	13.87%	Reversed	15.41%
2019/20	2002	13	Reversed	1.92%	Reversed	1.64%
2020/21	3305	9	Reversed	1.75%	Standard	2.42%
2021/22	4781	7	Standard	15.29%	Standard	18.76%
2022/23	3876	8	Standard	5.11%	Standard	5.87%
Average	5529.1	6.50	S: 12; R: 8	8.06%	S: 12; R: 8	8.76%

[4] L. Csató, Group draw with unknown qualified teams: A lesson from 2022 FIFA World Cup, *International Journal of Sports Science & Coaching*, in press, DOI: 10.1177/17479541221108799 (2022)

[5] L. Csató, A note on the UEFA Champions League Round of 16 draw, Manuscript, DOI: 10.48550/arXiv.2210.15555 (2022)

[6] FIBA, Procedure for FIBA Basketball World Cup 2019 Draw, https://www.fiba.basketball/basketballworldcup/2019/news/procedure-for-fiba-basketball-world-cup-2019-draw (2019)

[7] J. Guyon, Rethinking the FIFA World Cup™ final draw, Manuscript, DOI: 10.2139/ssrn.2424376 (2014)

[8] J. Guyon, Rethinking the FIFA World Cup™ final draw, *Journal of Quantitative Analysis in Sports*, **11**(3):169–182 (2015)

[9] J. Guyon, Ligue des champions : pourquoi le PSG a presque une chance sur trois de rencontrer Chelsea, *Le Monde*, 10 December, https://www.lemonde.fr/ligue-des-champions/article/2017/12/10/ligue-des-champions-pourquoi-le-psg-a-une-chance-sur-trois-de-rencontrer-chelsea_5227638_1616944.html (2017)

[10] J. Guyon, Pourquoi la Coupe du monde est plus équitable cette année. *The Conversation*, 13 June, https://theconversation.com/pourquoi-la-coupe-du-monde-est-plus-equitable-cette-annee-97948 (2018)

[11] J. Guyon, Champions League last-16 draw probabilities: Why Chelsea are more likely to get Barcelona – and what fates await Liverpool, Man City and Tottenham, *FourFourTwo*, 12 December, https://www.fourfourtwo.com/features/champions-league-last-16-draw-probabilities-liverpool-chelsea-tottenham-man-city-real-madrid-barcelona (2019)

[12] J. Guyon, Ligue des champions : fallait-il annuler complètement le résultat du premier tirage ?, *Le Monde*, 14 December, https://www.lemonde.fr/sport/article/2021/12/14/ligue-des-champions-fallait-il-annuler-completement-le-resultat-du-premier-tirage_6106012_3242.html (2021)

[13] J. Guyon, Ligue des champions : le Real Madrid et Chelsea, adversaires les plus probables du PSG et de Lille, *Le Monde*, 10 December, https://www.lemonde.fr/sport/article/2021/12/10/ligue-des-champions-real-madrid-et-chelsea-adversaires-les-plus-probables-du-psg-et-de-lille_6105478_3242.html (2021)

[14] J. Guyon, Ligue des champions : le Bayern Munich, adversaire le plus probable du PSG en huitièmes de finale. *Le Monde*, 7 November, https://www.lemonde.fr/sport/article/2022/11/07/ligue-des-champions-le-bayern-munich-adversaire-le-plus-probable-du-psg-en-huitiemes-de-finale_6148779_3242.html (2022)

[15] J. Haigh, *Mathematics in Everyday Life*, Springer Nature, Cham, Switzerland, second edition (2019)

[16] H. Kiesl, Match me if you can. Mathematische Gedanken zur Champions-League-Achtelfinalauslosung, *Mitteilungen der Deutschen Mathematiker-Vereinigung*, **21**(2):84–88 (2013)

[17] S. Klössner and M. Becker, Odd odds: The UEFA Champions League Round of 16 draw. *Journal of Quantitative Analysis in Sports*, **9**(3):249–270 (2013)

[18] G. O. Roberts and J. S. Rosenthal, Football group draw probabilities and corrections, Manuscript, DOI: 10.48550/arXiv.2205.06578 (2023)

[19] UEFA, FIFA World Cup 2022 qualifying draw procedure, https://www.uefa.com/MultimediaFiles/Download/competitions/WorldCup/02/64/22/19/2642219_DOWNLOAD.pdf (2020)

[20] UEFA, UEFA Nations League 2022/23 – league phase draw procedure, https://editorial.uefa.com/resources/026f-13c241515097-67a9c87ed1b2-1000/unl_2022-23_league_phase_draw_procedure_en.pdf (2021)

[21] UEFA (2022). UEFA EURO 2024 Qualifying Draw Procedure, https://editorial.uefa.com/resources/0279-1627b29793e1-ffbe5a3c77a1-1000/08.01.00_euro_2024_qualifying_draw_procedure_en_20220920115210.pdf (2022)

[22] M. Wright, OR analysis of sporting rules – A survey. *European Journal of Operational Research*, **232**(1):1–8 (2014)

Two-sided Convexity Testing with Certificates

Adrian Dumitrescu

Algoresearch L.L.C.,
Milwaukee, WI 53217, USA
ad.dumitrescu@algoresearch.org

Abstract: We revisit the problem of property testing for convex position for point sets in $\mathbb{R}^d$. Our results draw from previous ideas of Czumaj, Sohler, and Ziegler (ESA 2000). First, the algorithm is redesigned and its analysis is revised for correctness. Second, its functionality is expanded by (i) exhibiting both negative and positive certificates along with the convexity determination, and (ii) significantly extending the input range for moderate and higher dimensions.

The behavior of the randomized tester is as follows: (i) if P is in convex position, it accepts; (ii) if P is far from convex position, with probability at least $2/3$, it rejects and outputs a $(d+2)$-point witness of non-convexity as a negative certificate; (iiii) if P is close to convex position, with probability at least $2/3$, it accepts and outputs an approximation of the largest subset in convex position. The algorithm examines a sublinear number of points and runs in subquadratic time for every dimension d (and is faster in low dimensions).

Keywords: property testing, convex position, approximation algorithm, randomized algorithm

1 Introduction

A set of points in the d-dimensional space $\mathbb{R}^d$ is said to be: (i) in *general position* if any at most $d+1$ points are *affinely independent*; and (ii) in *convex position* if none of the points lies in the convex hull of the other points. It is known that every set of n points in general position in the plane contains $(1-o(1))\log n$ points in convex position, and this bound is tight up to lower-order terms [10, 24]. For $d \geq 3$, by the Erdős–Szekeres theorem, every set of n points in general position in $\mathbb{R}^d$ contains $\Omega(\log n)$ points in convex position (it suffices to find points whose projections onto a generic plane are in convex position). On the other hand, for every fixed $d \geq 2$, Károlyi and Valtr [15] and Valtr [25] constructed n-element sets in general position in $\mathbb{R}^d$ in which no more than $O(\log^{d-1} n)$ points are in convex position. A recent result of Pohoata and Zakharov [21] shows that $2^{o(n)}$ points in $\mathbb{R}^d$, $d \geq 3$, already contain n points in convex position.

Given a point set in general position in $\mathbb{R}^d$, the problem of computing a maximum-size subset in convex position can be solved in polynomial time for $d = 2$ by the dynamic programming algorithm of Chvátal and Klincsek [4]; their algorithm runs in $\mathcal{O}(n^3)$ time. In contrast, the general problem in $\mathbb{R}^d$ was shown to be NP-complete for every $d \geq 3$ by Giannopoulos, Knauer, and Werner [12], and moreover, no approximation algorithm is known.

Throughout this paper we assume (in a standard fashion) that our input set is a point set in $\mathbb{R}^d$ in *general position*. The complexity of computing the convex hull of n points in $\mathbb{R}^d$ is summarized in the following result of Chazelle; see also [1, 23]. Let P be a set of n points in $\mathbb{R}^d$, where d is considered constant.

Theorem 1 (Chazelle [3]) *Given P, the convex hull of P can be computed in $\mathcal{O}(n \log n + n^{\lfloor d/2 \rfloor})$ time using $\mathcal{O}(n^{\lfloor d/2 \rfloor})$ space, which is asymptotically worst-case optimal.*

It is known that the number of faces, f, of the output polytope is $\Theta(n^{\lfloor d/2 \rfloor})$ in the worst case [19] (a huge number). On the other hand, a result of Chan shows that the set of extreme points of a set of n points in $\mathbb{R}^d$ can be computed in subquadratic time and essentially faster when their number h is small.

Theorem 2 (Chan [2]) *Given P, the h extreme points of P can be computed in time*

$$T(n,h) = \mathcal{O}\left(n \log^{\mathcal{O}(1)} h + (nh)^{\frac{\lfloor d/2 \rfloor}{\lceil d/2 \rceil + 1}} \log^{\mathcal{O}(1)} n\right). \tag{1}$$

Taking $n = h$ in the above expression yields a time that suffices for testing whether a set of n points is in convex position. From the other direction, it is conjectured that the problem of testing whether a set P is in convex position is asymptotically as hard as the problem of computing all extreme points of P [5].

Corollary 3 (Chan [2]) *Given P, determining whether P is in convex position can be done in time* $T(n,n) = \mathcal{O}\left(n^{\frac{2\lfloor d/2 \rfloor}{\lceil d/2 \rceil + 1}} \log^{\mathcal{O}(1)} n\right)$.

In *property testing* one is concerned with the design of faster algorithms for approximate decision making [13]. In this scenario, instead of determining whether an input has a specific property, one determines if the input is *far* or perhaps *close* from satisfying that property. Such approximate decisions, usually involving random sampling or shortcuts in the computation, may be valuable in settings in which an exact decision is infeasible or just more expensive. For example, one may be interested in determining, given an input point set, how far it stands from being in convex position without needing to spend all resources that would be required for computing the convex hull of the respective set. Such a tool is obviously useful in the general area of testing properties of geometric objects and visual images for distinguishing a convex shape among other shapes.

The goal of *property testing* is to develop efficient *property testers*. Ideally, such a tester makes a sublinear number of queries of the input set (i.e., it does not look at all the input). However, this does not mean — even for the ideal case — that the tester runs in time that is sublinear in the size of the input; in fact, it often doesn't. Moreover, if the tester is also required to return a possibly large subset of the input set (depending on the outcome) as a certificate, then its time requirements may be further increased.

Here we focus on the testing of *convex position*. As in the context of randomized algorithms, approximately deciding means returning the correct answer with some confidence, specifically with probability at least 2/3 (as described below), see, e.g., [18]; however, the 2/3 threshold is not set in stone. For instance, in regard to the previous point on running time, it is worth noting that already for the plane ($d = 2$), testing for convex position by the algorithm in [5] takes $\mathcal{O}(n^{2/3}\varepsilon^{-1/3} \log(n/\varepsilon))$, which is $\Theta(n \log n)$ if $\varepsilon = \Theta(1/n)$; running times in higher dimensions are even higher.

Testing algorithms may use samples of different sizes. Some intuition is as follows. Suppose that the input is far from convex position; the algorithm is likely to reject on large samples (the larger the sample, the easier it will be to find that out), and is likely to accept on small samples (the smaller the sample, the easier the algorithm will be fooled). On the other hand, if the input is close to convex position, the smaller the sample, the easier it will be for the algorithm to accept.

A key distinction with regard to the action (accept or reject) is that closeness must fit the goal, i.e., far and close need to be quantified appropriately. As it turns out, rejecting an input that is far from convex position is relatively insensitive to the distance from convex position. However, when accepting an input that is close to convex position, the input must be really close.

1.1 Preliminaries

Definitions and notations. Let $0 < \varepsilon < 1/2$. A set P of n points is *ε-far* from convex position if there is no set $X \subset P$ of size at most εn such that $P \setminus X$ is in convex position. Otherwise, i.e., if there is a set $X \subset P$ of size at most εn such that $P \setminus X$ is in convex position, P is *ε-close* to convex position. See Fig. 1.

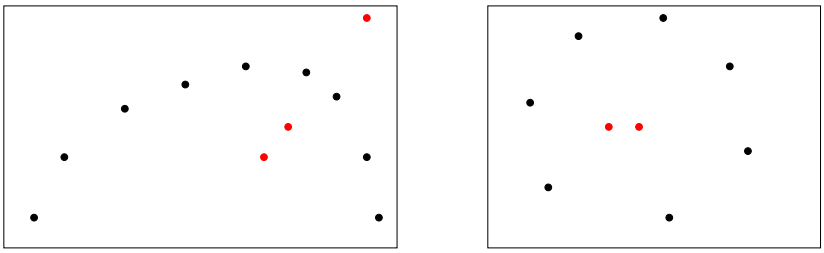


Figure 1: A 12-point set that is 1/4-close to convex position (left), and a 9-point set that is 2/9-close to convex position (right). Both sets are 1/5-far from convex position.

Here we use the convention that the approximation ratio of an algorithm is < 1 for a maximization problem and > 1 for a minimization problem (as in [26].) Unless specified otherwise, all logarithms are in base 2. For a set $W \subset \mathbb{R}^d$, its *interior* is denoted by $\mathring{W}$.

Nonconvexity certificates. By the well-known Carathéodory Theorem; see, e.g., [17, p. 6], if X is finite point set in $\mathbb{R}^d$, every non-extreme point of X can be expressed as a convex combination of at most $d+1$ points in X. This means that every point set that is not in convex position contains $d+2$ points that are not in convex position. We will further assume that Chan's algorithm for testing of convex position outputs such a tuple when the input is not in convex position.

The convex position tester of Czumaj, Sohler, and Ziegler. The convex position tester of Czumaj et al. [5] draws a random sample of the input set and makes a decision based on the convexity of this sample. The algorithm is set up to work in $\mathbb{R}^d$, for any dimension d. The tester accepts every point set in convex position, and rejects every point set that is ε-far from convex position with probability at least 2/3. If the input is not in convex position and is not ε-far from convex position, the outcome of the algorithm can go either way (i.e., there is no specified action for the situation in-between). Most of the technical justification is unpublished; for the present time, it can be found online [6]. The authors present two testers for convex position: `Convex-A` and `Convex-B`, see [5, p. 161].

Its *query complexity*, i.e., the number of points requested from an *oracle* to perform the testing, is $\mathcal{O}(n^{d/(d+1)}\varepsilon^{-1/(d+1)})$, which is shown by the authors to be optimal [6]. The corresponding *running time*, however, strongly depends on the dimension. For instance, it is $\mathcal{O}(n^{2/3}\varepsilon^{-1/3}\log(n/\varepsilon))$ for $d = 2$ and $\mathcal{O}(n^{3/4}\varepsilon^{-1/4}\log(n/\varepsilon))$ for $d = 3$; and subquadratic in any dimension d.

Unfortunately, the convex position tester of Czumaj et al. [5] suffers from both structural and performance issues as explained below. One issue is an unreasonable dependence of the tester `Convex-A` of the input parameter ε; another is an incorrect setting of the sample size of the tester `Convex-B`; a third concerns a technical lemma that needs correction. Here we fix these problems and obtain the first functional tester. Moreover, its functionality is expanded by including positive certificates. Our paper is self-contained with all needed proofs included.

(i) The sample size used by tester `Convex-A` is

$$s = 36 \cdot n^{\frac{d}{d+1}}\varepsilon^{-\frac{1}{d+1}}.$$

Since $s \leq n$ is a prerequisite for using the tester, this imposes the restriction $36^{d+1} \leq \varepsilon n$; equivalently, $\varepsilon \geq 36^{d+1}/n$. Since $\varepsilon < 1$, this implies $n > 36^{d+1}$. This requirement makes the tester impractical even for moderate values of d. For instance, if $d = 20$, tester `Convex-A` can only test sets with $n > 4.8 \cdot 10^{32}$ points. Similarly, if $d = 50$, tester `Convex-A` can only test sets with $n > 2.3 \cdot 10^{79}$ points, which is approximately the number of atoms in the observable universe. Arguably, such applications, if any, are rare. As such, the tester isn't functional in the range $d \geq 50$. In contrast, our Algorithm `Convex-` in Subsection 2.1 is only subject to the very modest restriction $\varepsilon \geq (d+1)/n$. Similarly, our Algorithm `Convex+` in Subsection 2.2 is subject to very modest restrictions.

(ii) Another issue is in regard to the correctness of the tester `Convex-B` in view of the sample size $s = 4/\varepsilon$ used by the tester. Suppose that $d = 4$ and the input is an n-element point set that is ε-far from convex position for a constant ε, say $\varepsilon = 1/4$, but not for a larger ε. By the optimality of the testing sample s mentioned above, it is required that $s = \Omega(n^{4/5}\varepsilon^{-1/5})$. For $s = 4/\varepsilon$, this implies $\varepsilon = O(1/n)$, which does not hold for large n. The tester `Convex-B` is therefore incorrect (its output is incorrect most of the time for the input described above and many others).

(iii) A third issue is the correctness of Lemma 3.4 in [6], discussed in Section A. Our Lemma 6 is proposed as a replacement.

Our results. We revisit the problem of property testing for convex position for point sets in $\mathbb{R}^d$. Our results draw from previous design and ideas of Czumaj, Sohler, and Ziegler (ESA 2000). First, the algorithm is redesigned and its analysis is revised for correctness. Second, its functionality is expanded by (i) exhibiting both negative and positive certificates along with the convexity determination, and (ii) significantly extending the input range for moderate and higher dimensions. The tester is implemented by two procedures: `Convex-` and `Convex+`.

The behavior of Algorithm `Convex-` can be summarized as follows. Let $0 < \varepsilon < 1$ be an input parameter.

1. If P is in convex position, the algorithm accepts P.
2. If P is ε-far from convex position, with probability at least $2/3$ the algorithm rejects P and outputs a $(d+2)$-point witness of non-convexity (as a negative certificate).

The behavior of Algorithm `Convex+` can be summarized as follows. Let $0 < \varepsilon < 1$ be an input parameter, and $0 < \delta \leq 1/2$ be an adjustable parameter. Here we work with $\delta = 0.1$.

1. If P is in convex position, the algorithm accepts P.
2. If P is ε-close to convex position for some $\varepsilon > 0$ that satisfies $n^{-1} \leq \varepsilon \leq n^{\delta-1}$, with probability at least $2/3$ the algorithm accepts P and outputs a $1/(6n^{\delta})$-approximation of the largest subset in convex position (as a positive certificate).

Related work. A seminal article in the area of property testing is due to Ergün et al. [11]. Besides testing for convex position, testing for other geometric properties has been considered in [5]: pairwise disjointness of a set of generic bodies, disjointness of two polytopes, and Euclidean minimum spanning tree verification. A continuation of the work in [5] appears in [7]. A more recent article on property testing for point sets in the plane is due to Han et al. [14]. A recent monograph dedicated to the general subject of property testing is [13]. The topic of property testing, including testing for convex position, is also addressed in a recent book by Eppstein [9].

2 An enhanced functionality tester for convex position

The tester is implemented by two procedures: Algorithm `Convex-` (in Subsection 2.1) and Algorithm `Convex+` (in Subsection 2.2). The two procedures may be run independently of each other. The goal of Algorithm `Convex-` is rejecting point sets that are far from convex position; whereas that of Algorithm `Convex+` is accepting point sets that are close to convex position. Each algorithm exhibits a suitable certificate along with its probabilistic determination. While the decision is randomized, the certificates produced are indisputable, i.e., a negative certificate is always a $(d+2)$-point set that is not in convex position, and a positive certificate output by Algorithm `Convex+` is always a $1/(6n^{\delta})$-approximation of the largest subset in convex position.

Common tools. A randomized algorithm for generating a random s-set for a given s, $1 \leq s \leq n$, in $\mathcal{O}(s \log s)$ time (and $\mathcal{O}(s)$ expected time) from [20, Ch. 4], can be used to implement random sample selection. Alternatively, a linear-time algorithm for the same task from [22, Sec 5.2] can also be used.

2.1 Negative testing: Algorithm `Convex-`

Several constraints among the input parameters need to be respected usually for technical reasons. In particular, it is assumed that (note that these constraints are very mild):

- $n \geq 2^{10}$, this is needed in the proof of Lemma 6.
- $n \geq 32(d+1)$, this ensures that $\ell \leq n/32$ when using Lemma 6.
- $\varepsilon \geq \frac{10(d+1)}{n}$, this ensures that $k \geq 10$ in Step 1; compare this to the constraint $\varepsilon \geq 36^{d+1}/n$ in tester `Convex-A` that restricts its use to low dimensions.
- $\varepsilon \leq \frac{d-1}{2d}$, this ensures $\frac{(1-\varepsilon)}{d+1} \geq \frac{1}{2d}$ in the analysis.

Algorithm `Convex-`

Step 1: Let $k = \lfloor \frac{\varepsilon n}{d+1} \rfloor$, $\ell = d+1$, $s_0 = \ell + \frac{n-\ell}{(2k)^{1/\ell}}$, and $s = \lceil s_0 \rceil$. Repeat Step 2 and Step 3 in succession up to 22 times.

Step 2: Randomly select a subset $S \subset P$ of size s, with all s-subsets being equally likely.

Step 3: Test S for convex position using Chan's algorithm. If S is not in convex position, outputs a $(d+2)$-point witness of non-convexity and reject P. Otherwise go to Step 2 for the next repetition.

Step 4: If all 22 samples were determined to be in convex position, accept P.

Time analysis. It is easily verified that the setting for s in Step 1 yields

$$s = \Theta\left(n^{\frac{d}{d+1}} \varepsilon^{-\frac{1}{d+1}}\right).$$

This is in accordance with the choice of the sample size for Algorithm `Convex-A` in [5]. As such, the runtime of Algorithm `Convex-` is

$$\begin{aligned} T(s,s) &= \mathcal{O}\left(T\left(n^{\frac{d}{d+1}} \varepsilon^{-\frac{1}{d+1}}, n^{\frac{d}{d+1}} \varepsilon^{-\frac{1}{d+1}}\right)\right) \\ &= \mathcal{O}\left(n^{\frac{d}{d+1} \cdot \frac{2\lfloor d/2 \rfloor}{\lfloor d/2 \rfloor + 1}} \cdot \varepsilon^{-\frac{1}{d+1} \cdot \frac{2\lfloor d/2 \rfloor}{\lfloor d/2 \rfloor + 1}} \cdot \log^{\mathcal{O}(1)}(n/\varepsilon)\right). \end{aligned}$$

Since $\varepsilon = \Omega(1/n)$, the above expression becomes

$$T(s,s) = \mathcal{O}\left(T(n,n)\right) = \mathcal{O}\left(n^{\frac{2\lfloor d/2 \rfloor}{\lfloor d/2 \rfloor + 1}} \log^{\mathcal{O}(1)} n\right) = o(n^2), \text{ for every } n \text{ and } \varepsilon.$$

This can be also seen directly: since $s \leq n$, $T(s,s) \leq T(n,n) = o(n^2)$.

Rejecting the input with probability $\geq 2/3$. Assume that P is ε-far from convex position. We show that with probability at least 2/3, Algorithm `Convex-` rejects the input in step 3 and outputs a suitable $(d+2)$-point witness. We first recall the following lemmas (analogous to Lemma 3.1 and 3.2 from [6]), slightly rewritten here for convenience.

Lemma 4 (An earlier version in [6]). *Let $P \subset \mathbb{R}^d$ be a set of n points that is not in convex position and $p \in P$ be an interior point. Then there exist points $p_1, \ldots, p_d \in P$ and $U \subset P \setminus \{p_1, \ldots, p_d, p\}$ with $|U| \geq \frac{n-1}{d+1}$ such that $\{p_1, \ldots, p_d, p\} \cup \{q\}$ is not in convex position for every $q \in U$; more precisely, p is an interior point in the simplex $\Delta(p_1, \ldots, p_d, q)$ for every $q \in U$.*

Proof: Since $p \in P$ is an interior point, by Caratheodory's Theorem [17, p. 6] and by the general position assumption, there exists a set $W \subset P$ of size $d+1$ such that $p \in \mathring{W}$. See Fig. 2.

Denote by W_i, $i = 1, \ldots, d+1$, the d subsets of W of size d. We show that one of the subsets W_i of W satisfies the requirement in the lemma. We may assume without loss of generality that $p =$

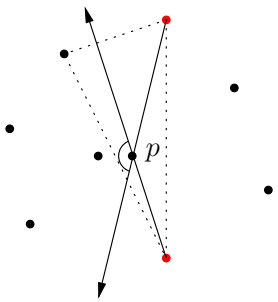


Figure 2: P is a set of 9 points in the plane. The cone determined by the two red points contains $4 \geq 8/3$ points in P.

$(0, \ldots, 0)$. We partition $\mathbb{R}^d$ into $d+1$ cones as follows. Let W_i^-, $i = 1, \ldots, d+1$, denote the set of points $\{(-x_1, \ldots, -x_d) \colon (x_1, \ldots, x_d) \in W_i\}$. The conic combination of the point vectors in the set W_i^- defines a cone C_i, $i = 1, \ldots, d+1$. The union of these cones cover $\mathbb{R}^d$. Thus there is a cone C_j, $1 \leq j \leq d+1$, that contains at least $\frac{n-1}{d+1}$ points in P. Observe that for every $q \in P \cap C_j$ we have $p \in (W_j \cup \{q\})^\circ$. Consequently, one can set $\{p_1, \ldots, p_d\} = W_j$ to conclude the proof. □

The following lemma applies to point sets that are far from convex position. The sets W_i and U_i constructed in the lemma are fixed before the samplings and are only used in the algorithm analysis.

Lemma 5 (An earlier version in [6]). *Let $P \subset \mathbb{R}^d$ be a set of n points that is ε-far from convex position and let $k = \lfloor \frac{\varepsilon n}{d+1} \rfloor$. Then there exist sets $W_i, U_i \subset P$ for $1 \leq i \leq k$, such that the following conditions are satisfied:*

(i) $|W_i| = d+1$ *for* $1 \leq i \leq k$,

(ii) $W_i \cap W_j = \emptyset$ *for all* $1 \leq i < j \leq k$,

(iii) $W_i \cap U_i = \emptyset$ *for* $1 \leq i \leq k$,

(iv) $W_i \cup \{q\}$ *is not in convex position for every* $q \in U_i$, *and*

(v) $|U_i| \geq \frac{n}{d+1} - k$ *for* $1 \leq i \leq k$. *In particular,* $|U_i| \geq \frac{(1-\varepsilon)n}{d+1}$.

Proof: We construct point sets $P_1, P_2, \ldots, P_k$ iteratively. We initially set $P_1 := P$ and then iteratively find $W_i \subset P_i$ and set $P_{i+1} := P_i \setminus W_i$ for $i = 1, \ldots, k$. By construction the sets W_i are pairwise disjoint, as required. Assuming that $|W_i| = d+1$ for $1 \leq i \leq k$, implies that

$$|P_i| = n - (d+1)(i-1) \geq n - (d+1)(k-1) > n - (d+1)\frac{\varepsilon n}{d+1} = (1-\varepsilon)n.$$

By the assumption in the lemma, P_i cannot be in convex position. By Lemma 4 there exist $p_1, \ldots, p_d, p \in P_i$ and $U_i \subset P_i \setminus \{p_1, \ldots, p_d, p\}$ with

$$\begin{aligned} |U_i| &\geq \frac{|P_i| - 1}{d+1} \geq \frac{n - (d+1)(i-1) - 1}{d+1} \geq \frac{n - (d+1)(k-1) - 1}{d+1} \\ &> \frac{n}{d+1} - k = \frac{n}{d+1} - \frac{\varepsilon n}{d+1} = \frac{(1-\varepsilon)n}{d+1}, \end{aligned}$$

such that p is an interior point in the simplex $\Delta p_1, \ldots, p_d, q$ for every $q \in U_i$. Let $W_i := \{p_1, \ldots, p_d, p\}$ and observe that $W_i \cap U_i = \emptyset$. Note that all properties in the lemma have been verified. □

We also need another lemma suggested by Czumaj et al. [6]. Here we include a proof that follows the ideas of the original proof, however, it is revised for correctness and for a slightly restricted range of the parameters that suffices for our purposes. More details can be found in Section A.

Lemma 6 (An earlier version in [6]). *Let Ω be a set of size n and $W_1, W_2, \ldots, W_k \subset \Omega$ be k pairwise disjoint subsets of Ω of size ℓ, where $k \geq 10$ and $3 \leq \ell \leq n/32$. Let s be a positive integer such that $\ell + \frac{n-\ell}{(2k)^{1/\ell}} \leq s \leq n$ and $S \subset \Omega$ be a subset of Ω of size s chosen uniformly at random. Then*

$$\text{Prob}(\exists i \leq k\colon (W_i \subset S)) \geq \frac{1}{4}.$$

PROOF: Observe that $k\ell \leq n$, hence $k \leq n/\ell$. Let s_0 be the real number defined as follows:

$$s_0 = \ell + \frac{n-\ell}{(2k)^{1/\ell}}, \text{ or } k\left(\frac{s_0-\ell}{n-\ell}\right)^{\ell} = \frac{1}{2}, \tag{2}$$

and note that $\ell < s_0 < n$. Indeed, the lower bound is clear and the upper bound $s_0 < n$ is equivalent to $(2k)^{1/\ell} > 1$ which is obvious. We first prove that

$$s_0 \geq 3\ell \log k. \tag{3}$$

It suffices to show that $n - \ell \geq 3\ell(2k)^{1/\ell} \log k$, or, since $\ell \leq n/32$, that $3\ell(2k)^{1/\ell} \log k \leq \frac{31n}{32}$. We have

$$3\ell(2k)^{1/\ell} \log k \leq 3\ell \left(\frac{2n}{\ell}\right)^{1/\ell} \log\left(\frac{2n}{\ell}\right) \leq \frac{31n}{32}.$$

Indeed, a standard verification shows that the function

$$f(x) = 3x\left(\frac{2n}{x}\right)^{1/x} \log\left(\frac{2n}{x}\right), x \in \left[3, \frac{n}{32}\right],$$

where $n \geq 2^{10}$, attains it maximum at $x = n/32$, thus

$$\begin{aligned} f(x) \leq f\left(\frac{n}{32}\right) &= 3 \cdot \frac{n}{32} \cdot \left(\frac{2n}{n/32}\right)^{32/n} \log\left(\frac{2n}{n/32}\right) \\ &= \frac{3n}{32} \cdot 64^{32/n} \cdot \log 64 \leq \frac{18n}{32} \cdot \frac{5}{4} \leq \frac{31n}{32}. \end{aligned}$$

This concludes the proof of (3) and we next focus on the inequality in the lemma.

Since the probability in question increases as the sample size s grows, it suffices to prove the inequality for $s = \lceil s_0 \rceil$. Observe that $\ell + 1 \leq s \leq n$. By the Boole-Bonferoni inequality—see, e.g., [16, Ch. 2], we have

$$\text{Prob}(\exists i \leq k\colon (W_i \subset S)) \geq \sum_{i=1}^{k} \text{Prob}(W_i \subset S) - \sum_{1 \leq i < j \leq k} \text{Prob}((W_i \cup W_j) \subset S). \tag{4}$$

It is easily verified that

$$\begin{aligned} \text{Prob}(W_i \subset S) &= \frac{\binom{n-\ell}{s-\ell}}{\binom{n}{s}} = \frac{(n-\ell)!}{(s-\ell)!(n-s)!} \cdot \frac{s!(n-s)!}{n!} \\ &= \frac{(n-\ell)!s!}{n!(s-\ell)!} = \prod_{r=0}^{\ell-1} \frac{s-r}{n-r}, \text{ and} \\ \text{Prob}((W_i \cup W_j) \subset S) &= \frac{\binom{n-2\ell}{s-2\ell}}{\binom{n}{s}} = \prod_{r=0}^{2\ell-1} \frac{s-r}{n-r} \\ &= \prod_{r=0}^{\ell-1} \frac{s-r}{n-r} \cdot \prod_{r=0}^{\ell-1} \frac{(s-\ell)-r}{(n-\ell)-r}, \text{ for } 1 \leq i < j \leq k. \end{aligned}$$

Substituting these into Inequality (4) and finally using (2) yields

$$\begin{aligned}
\text{Prob}(\exists i \le k\colon (W_i \subset S)) &\ge k \cdot \prod_{r=0}^{\ell-1} \frac{s-r}{n-r} - \binom{k}{2} \cdot \prod_{r=0}^{\ell-1} \frac{s-r}{n-r} \cdot \prod_{r=0}^{\ell-1} \frac{(s-\ell)-r}{(n-\ell)-r} \\
&= k \cdot \prod_{r=0}^{\ell-1} \frac{s-r}{n-r} \left(1 - \frac{k-1}{2} \cdot \prod_{r=0}^{\ell-1} \frac{(s-\ell)-r}{(n-\ell)-r}\right) \\
&\ge k \cdot \prod_{r=0}^{\ell-1} \frac{s-\ell}{n-\ell} \cdot \left(1 - \frac{k}{2} \cdot \prod_{r=0}^{\ell-1} \frac{s-\ell}{n-\ell}\right) \\
&= k \cdot \left(\frac{s-\ell}{n-\ell}\right)^{\ell} \cdot \left(1 - \frac{k}{2} \cdot \left(\frac{s-\ell}{n-\ell}\right)^{\ell}\right).
\end{aligned}$$

Let

$$F_1 = k \cdot \left(\frac{s-\ell}{n-\ell}\right)^{\ell} \text{ and } F_2 = 1 - \frac{k}{2} \cdot \left(\frac{s-\ell}{n-\ell}\right)^{\ell}.$$

It suffices to show that $F_1 \ge \frac{1}{2}$ and $F_2 \ge \frac{1}{2}$. For the first inequality, we have

$$F_1 = k \cdot \left(\frac{s-\ell}{n-\ell}\right)^{\ell} \ge k \cdot \left(\frac{s_0-\ell}{n-\ell}\right)^{\ell} = \frac{1}{2}. \tag{5}$$

For the second, recall that $0 \le s - s_0 < 1$ and $s_0 \ge 6\ell \ge 3\ell$ by (3). Applying the standard inequality $1 + x \le \mathrm{e}^x$ for $0 \le x \le 1/2$ yields:

$$\left(\frac{s-\ell}{s_0-\ell}\right)^{\ell} = \left(1 + \frac{s-s_0}{s_0-\ell}\right)^{\ell} \le \left(1 + \frac{1}{2\ell}\right)^{\ell} \le \exp(0.5) \le 2. \tag{6}$$

Using (6) and (2) once again yields

$$\begin{aligned}
F_2 &= 1 - \frac{k}{2} \cdot \left(\frac{s-\ell}{n-\ell}\right)^{\ell} = 1 - \left(\frac{s-\ell}{s_0-\ell}\right)^{\ell} \cdot \frac{k}{2} \cdot \left(\frac{s_0-\ell}{n-\ell}\right)^{\ell} \\
&\ge 1 - 2 \cdot \frac{k}{2} \cdot \left(\frac{s_0-\ell}{n-\ell}\right)^{\ell} = 1 - k \cdot \left(\frac{s_0-\ell}{n-\ell}\right)^{\ell} = \frac{1}{2}.
\end{aligned} \tag{7}$$

Consequently, we have

$$\text{Prob}(\exists i \le k\colon (W_i \subset S)) \ge F_1 \cdot F_2 \ge \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4},$$

as required. □

Let $k = \lfloor \frac{\varepsilon n}{d+1} \rfloor$, $\ell = d + 1$, and recall that Algorithm `Convex-` sets $s = \lceil s_0 \rceil$, where s_0 is given by Equation (2).

We next prove that the algorithm finds the sample S not convex with probability $\ge 1/20$ in each of the 22 repetitions in Step 2 and Step 3. Consider one execution of Step 2 and Step 3. For a fixed $i \le k$, let F_i be the event that $S \cap U_i = \emptyset$. By Lemma 5, we have $|U_i| \ge \frac{(1-\varepsilon)n}{d+1} \ge \frac{n}{2d}$. Observe that

$$\left(1 - \frac{1}{2d}\right)^{d+1} \le \frac{2}{3}, \text{ for } d \ge 2.$$

By (3) we have $s \geq s_0 \geq 3\ell \log k$, thus (recall also that $k \geq 10$, which us used in the last inequality of the chain below)

$$\begin{aligned}
\mathrm{Prob}(F_i) = \mathrm{Prob}(S \cap U_i = \emptyset) &= \frac{\binom{n-|U_i|}{s}}{\binom{n}{s}} \\
&= \frac{(n-|U_i|)(n-|U_i|-1)\cdots(n-|U_i|-s+1)}{n(n-1)\cdots(n-s+1)} \leq \left(1-\frac{|U_i|}{n}\right)^s \\
&\leq \left(1-\frac{1}{2d}\right)^s \leq \left(1-\frac{1}{2d}\right)^{3\ell\log k} \\
&\leq \left(\frac{2}{3}\right)^{3\log k} \leq \frac{1}{5k}, \text{ for } i \in [k] \text{ and } d \geq 2.
\end{aligned}$$

Let E_1 be the event that $S \cap U_i \neq \emptyset$ for every $i \leq k$. By the union bound, we deduce that

$$\mathrm{Prob}(\overline{E_1}) \leq k \cdot \mathrm{Prob}(F_1) \leq \frac{1}{5}.$$

Let E_2 be the event that there exists $i \leq k$ such that $W_i \subset S$. We next verify that the inequality $\ell + \frac{n-\ell}{(2k)^{1/\ell}} \leq s \leq n$ specified in Lemma 6 holds. Indeed,

$$s = \lceil s_0 \rceil \geq s_0 = \ell + \frac{n-\ell}{(2k)^{1/\ell}},$$

and $s_0 < n$ as shown in the proof of Lemma 6, whence $s = \lceil s_0 \rceil \leq n$. Hence by Lemma 6 we have

$$\mathrm{Prob}(E_2) = \mathrm{Prob}(\exists i \leq k\colon (W_i \subset S)) \geq \frac{1}{4}.$$

Putting these bounds together yields

$$\begin{aligned}
\mathrm{Prob}(E_1 \cap E_2) &= 1 - \mathrm{Prob}(\overline{E_1} \cup \overline{E_2}) \geq 1 - \mathrm{Prob}(\overline{E_1}) - \mathrm{Prob}(\overline{E_2}) \\
&\geq 1 - \frac{1}{5} - (1 - \mathrm{Prob}(E_2)) = \mathrm{Prob}(E_2) - \frac{1}{5} \\
&\geq \frac{1}{4} - \frac{1}{5} = \frac{1}{20}.
\end{aligned}$$

Let E be the event that Algorithm `Convex-` finds the sample not convex in at least one of the 22 executions of Step 2 and Step 3. The 22 repetitions are independent events, thus

$$\mathrm{Prob}(E) \geq 1 - \left(1 - \frac{1}{20}\right)^{22} \geq \frac{2}{3}.$$

Thus with probability at least 2/3, Algorithm `Convex-` rejects the input, as required.

2.2 Positive testing: Algorithm `Convex+`

Assume for technical reasons that n is sufficiently large ($n \geq 1500$). Let $0 < \delta \leq 1/2$ be an adjustable parameter. Here we work with $\delta = 0.1$. Assume that P is ε-close to convex position for some $\varepsilon > 0$, where $n^{-1} \leq \varepsilon \leq n^{\delta-1}$; note, this means that P can be made convex by removing at most $\varepsilon n \leq n^{\delta}$ points.

Algorithm `Convex+`

Step 1: Randomly select a subset $S \subset P$ of size $s = \lceil 1/(6\varepsilon) \rceil$, with all s-subsets being equally likely.

Step 2: Test S for convex position using Chan's algorithm. If S is not in convex position, outputs a $(d+2)$-point witness of non-convexity and reject P. Otherwise output S as a subset in convex position and accept P.

Time analysis. The setting $s = \lceil 1/(6\varepsilon) \rceil$ in Step 1 yields that the runtime of Algorithm `Convex+` is

$$T(s,s) = \mathcal{O}\left(T(1/\varepsilon, 1/\varepsilon)\right) = \mathcal{O}\left(\varepsilon^{-\frac{2\lfloor d/2 \rfloor}{\lceil d/2 \rceil + 1}} \log^{\mathcal{O}(1)} 1/\varepsilon\right).$$

Since $\varepsilon = \Omega(1/n)$,

$$T(s,s) = \mathcal{O}\left(T(n,n)\right) = \mathcal{O}\left(n^{\frac{2\lfloor d/2 \rfloor}{\lceil d/2 \rceil + 1}} \log^{\mathcal{O}(1)} n\right) = o(n^2), \text{ for every } n \text{ and } \varepsilon.$$

Accepting the input with probability $\geq 2/3$. We next show that with probability at least $2/3$, Algorithm `Convex+` accepts P and outputs a subset of size $\lceil 1/(6\varepsilon) \rceil$ of P in convex position. By the assumption we can write $P = C \cup D$, where C is in convex position and $|D| \leq \varepsilon n =: t$. Recall that $s = \lceil 1/(6\varepsilon) \rceil$. Note that

$$st = \left\lceil \frac{1}{6\varepsilon} \right\rceil \cdot \varepsilon n \leq \frac{1}{6\varepsilon} \cdot \varepsilon n + \varepsilon n = \frac{n}{6} + \varepsilon n \leq \frac{100n}{595} \text{ for } n \geq 1500.$$

Indeed, $n \geq 1500 \implies n^{0.9} \geq 721 \implies \varepsilon \leq 1/n^{0.9} \leq 1/721$, for which the above inequality holds. In particular, we have $t \leq st \leq 100n/595$. We show that

$$\text{Prob}(S \cap D = \emptyset) = \text{Prob}(S \subseteq C) \geq \frac{2}{3}.$$

Applying the standard inequality $1 - x \geq \mathrm{e}^{-2x}$ for $0 \leq x \leq 1/2$ yields:

$$\begin{aligned}
\text{Prob}(S \subseteq C) = \frac{\binom{|C|}{s}}{\binom{n}{s}} &\geq \frac{\binom{n-t}{s}}{\binom{n}{s}} = \frac{(n-s)(n-s-1)\cdots(n-s-t+1)}{n(n-1)\cdots(n-t+1)} \\
&= \prod_{i=0}^{t-1} \left(1 - \frac{s}{n-i}\right) \geq \left(1 - \frac{s}{n-t+1}\right)^t \geq \exp\left(\frac{-2st}{n-t+1}\right) \\
&\geq \exp\left(\frac{-200}{495}\right) \geq \frac{2}{3},
\end{aligned}$$

as required. Hence with probability at least $2/3$, S is determined to be in convex position and output by the algorithm, as required. Let OPT denote the size of the largest convex subset of P. Since $\mathsf{OPT} \leq n$ and $\varepsilon n \leq n^\delta$, the approximation ratio of Algorithm `Convex+` is

$$\frac{s}{\mathsf{OPT}} \geq \frac{s}{n} = \left\lceil \frac{1}{6\varepsilon} \right\rceil \frac{1}{n} \geq \frac{1}{6\varepsilon n} \geq \frac{1}{6n^\delta}.$$

In particular, the ratio is at least $1/24$ for all $n \leq 10^6$.

3 Concluding remarks

Summary. We presented and analyzed a convexity-testing algorithm implemented by two procedures based on random sampling that has the following enhanced functionality:

1. For point sets that are ε-far from convex position, with probability $\geq 2/3$ the algorithm outputs a $(d+2)$-point witness of non-convexity as a negative certificate.
2. For point sets that are ε-close to convex position, with probability $\geq 2/3$ the algorithm outputs a $1/(6n^\delta)$-approximation of a maximum-size convex subset ($\delta = 0.1$). [Comment: The current fastest algorithm for computing the largest subset in convex position takes $\mathcal{O}(n^3)$ time for $d = 2$, see [4, 8]. In contrast, the problem of computing a largest subset of points in convex position is NP-complete for $d \geq 3$ [12], and moreover, no approximation algorithm is known.]
3. The input range for the tester is significantly extended — for moderate and higher dimensions — compared to the previous version in [5].

References

[1] Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars, *Computational Geometry*, 3rd edition, Springer, Heidelberg, 2008.

[2] Timothy M. Chan, Output-sensitive results on convex hulls, extreme points, and related problems, *Discrete & Computational Geometry* **16(4)** (1996), 369–387.

[3] Bernard Chazelle, An optimal convex hull algorithm in any fixed dimension, *Discrete & Computational Geometry* **10** (1993), 377–409.

[4] Vašek Chvátal and Gheza T. Klincsek, Finding largest convex subsets, *Congressus Numerantium*, **29** (1980), 453–460.

[5] Artur Czumaj, Christian Sohler, and Martin Ziegler, Property testing in computational geometry (extended abstract), in *Proc. 8th Annual European Symposium on Algorithms* (ESA 2000), Springer, Heidelberg, vol. 1879 of LNCS, pp. 155–166. `https://doi.org/10.1007/3-540-45253-2_15`.

[6] Artur Czumaj, Christian Sohler, and Martin Ziegler, Testing convex position, `https://www.researchgate.net/publication/228727099_Testing_Convex_Position`. Online manuscript (16 pages), accessed in April 2022.

[7] Artur Czumaj and Christian Sohler, Property testing with geometric queries, in *Proc. 9th Annual European Symposium on Algorithms* (ESA 2001), Springer, Heidelberg, vol. 2161 of LNCS, pp. 266–277. `https://doi.org/10.1007/3-540-44676-1_22`.

[8] Herbert Edelsbrunner and Leonidas J. Guibas, Topologically sweeping an arrangement, *Journal of Computer and System Sciences* **38(1)** (1989), 165–194.

[9] David Eppstein, *Forbidden Configurations in Discrete Geometry*, Cambridge University Press, 2018.

[10] Paul Erdős and György Szekeres, A combinatorial problem in geometry, *Compositio Mathematica* **2** (1935), 463–470.

[11] Funda Ergün, Sampath Kannan, Ravi S. Kumar, Ronitt Rubinfeld, and Mahesh Viswanathan, Spot-checkers, *Journal of Computer and System Sciences* **60(3)** (2000), 717–751.

[12] Panos Giannopoulos, Christian Knauer, and Daniel Werner, On the computational complexity of Erdős-Szekeres and related problems in $\mathbb{R}^3$, *Proc. 21st European Symposium on Algorithms*, vol. 8125 of LNCS (2013), pp. 541–552.

[13] Oded Goldreich, *Introduction to Property Testing*, Cambridge University Press, 2017.

[14] Jie Han, Yoshiharu Kohayakawa, Marcelo T. Sales, and Henrique Stagni, Property testing for point sets on the plane, *Proc. of Latin American Symposium on Theoretical Informatics* (LATIN 2018), Springer, vol. 10807 of LNCS, pp. 584–596.

[15] Gyula Károlyi and Pavel Valtr, Configurations in d-space without large subsets in convex position, *Discrete & Computational Geometry* **30(2)** (2003), 277–286.

[16] László Lovász, *Combinatorial Problems and Exercises*, 2nd edition, Elsevier, Amsterdam, 1993.

[17] Jiří Matoušek, *Lectures on Discrete Geometry*, Springer, New York, 2002.

[18] Michael Mitzenmacher and Eli Upfal, *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*, 2nd edition, Cambridge University Press, 2017.

[19] Peter McMullen, The maximal number of faces of a convex polytope. *Mathematika* **17** (1970), 179–184.

[20] Albert Nijenhuis and Herbert S. Wilf, *Combinatorial Algorithms*, 2nd edition, Academic Press, New York, 1978.

[21] Cosmin Pohoata and Dmitrii Zakharov, Convex polytopes from fewer points, manuscript, August 2022. Preprint available at `arXiv.org/abs/2208.04878`.

[22] Edward M. Reingold, Jurg Nievergelt, and Narsingh Deo, *Combinatorial Algorithms: Theory and Practice*, Prentice-Hall, New Jersey, 1977.

[23] Raimund Seidel, Convex hull computations, Chap. 26 in *Handbook of Discrete and Computational Geometry* (Jacob E. Goodman, Joseph O'Rourke, and Csaba D. Tóth, eds.), 3rd edition, CRC Press, Boca Raton, 2017, pp.1057–1092.

[24] Andrew Suk, On the Erdős-Szekeres convex polygon problem, *Journal of the American Mathematical Society* **30** (2017), 1047–1053.

[25] Pavel Valtr, Convex independent sets and 7-holes in restricted planar point sets, *Discrete & Computational Geometry* **7(2)** (1992), 135–152.

[26] David P. Williamson and David B. Shmoys, *The Design of Approximation Algorithms*, Cambridge University Press, 2011.

A Remarks on Lemma 3.4 in [6]

The following lemma is suggested in [6]. Here we argue why the lemma cannot be used as is.

Lemma 7 [6]. *Let Ω be an arbitrary set set of n elements. Let k and ℓ be arbitrary integers (possibly dependent on n) and let s be an arbitrary integer such that $s \geq 2n/(2k)^{1/\ell}$. Let $W_1, W_2, \ldots, W_k$ be arbitrary disjoint subsets of Ω each of size ℓ. Let W be a subset of Ω of size s which is chosen independently and uniformly at random. Then*

$$\text{Prob}(\exists j \in [k] \colon (W_j \subseteq W)) \geq \frac{1}{4}.$$

We make two points:

(i) The first point is minor: taking s as the smallest integer satisfying $s \geq 2n/(2k)^{1/\ell}$, namely $s = \lceil 2n/(2k)^{1/\ell} \rceil$ may result in an integer larger than n and thereby be infeasible. For example, the setting $n = 256$, $k = 8$, $\ell = 8$, yields $s = \lceil 2n/(2k)^{1/\ell} \rceil = 363 > 256$.

(ii) The second point requires attention. Reading through the first few lines of the proof suggests that one could take

$$s = \ell + \frac{n-\ell}{(2k)^{1/\ell}}, \text{ or } k\left(\frac{s-\ell}{n-\ell}\right)^{\ell} = \frac{1}{2}. \tag{8}$$

However, this value may be not an integer, and thereby be again infeasible. Suppose that one takes instead the ceiling in the expression of s:

$$s = \ell + \left\lceil \frac{n-\ell}{(2k)^{1/\ell}} \right\rceil. \tag{9}$$

For the above setting in (i), this yields $s = 8 + \left\lceil \frac{248}{(16)^{1/8}} \right\rceil = 8 + 176 = 184$. Then the two factors that appear in the calculation of the lower bound on the probability in question are

$$F_1 = k \cdot \left(\frac{s-\ell}{n-\ell}\right)^{\ell} = 8 \cdot \left(\frac{176}{248}\right)^{8} = 0.5147\ldots,$$
$$F_2 = 1 - k \cdot \left(\frac{s-\ell}{n-\ell}\right)^{\ell} = 1 - 8 \cdot \left(\frac{176}{248}\right)^{8} = 0.4852\ldots$$

It is now clear that $F_1 \cdot F_2 < \frac{1}{4}$. (Taking the floor does not work either.) The above example is not an exception, and this occurs whenever the value of s in (8) is not an integer, which happens most of the time.

Results on extremal graph theoretic questions for q-ary vectors

Koppány Encz[1]

Eötvös Loránd University
Budapest, Hungary
enczkoppany@gmail.com

Márton Marits[2]

Budapest University of Technology and Economics
Hungary
marits.marton@gmail.com

Benedek Váli[3]

University of Cambridge
UK
benedekvali@gmail.com

Máté Weisz[4]

University of Cambridge
UK
weisz.mate.barnabas@gmail.com

Abstract: A q-graph with e edges and n vertices is defined as an $e \times n$ matrix with entries from $\{0, \ldots, q\}$, such that each row of the matrix (called a q-edge) contains exactly two nonzero entries. If H is a q-graph, then H is said to contain an s-copy of the ordinary graph F, if a set S of q-edges can be selected from H such that their intersection graph is isomorphic to F, and for any vertex v of S and any two incident edges $e, f \in S$ the sum of the entries of e and f is at least s. The extremal number $\mathrm{ex}(n, F, q, s)$ is defined as the maximal number of edges in an n-vertex q-graph such that it does not contain contain an s-copy of the forbidden graph F.

In the present paper, we reduce the problem of finding $\mathrm{ex}(n, F, q, q+1)$ for even q to the case $q = 2$, and determine the asymptotics of $\mathrm{ex}(n, C_{2k+1}, q, q+1)$.

Keywords: q-ary vectors, extremal graph theory, Turán number

1 Introduction

In his early papers ([6] and [7]), Pál Turán established the foundations of the broad area in mathematics called extremal graph theory. The key concept of the topic, denoted by $\mathrm{ex}(n, F)$, is defined as the maximum number of edges an n-vertex graph may have without containing F as a subgraph. The asymptotics of $\mathrm{ex}(n, F)$ have been determined by Erdős, Stone and Simonovits in [2] and [1] for any non-bipartite F. Since then, several related problems for bipartite graphs have been resolved (for an exhaustive collection of relevant results, see [3]), but many questions still remain open.

Patkós, Tuza and Vizer, in pursuit of a generalization for Turán-problems, have introduced the notion of q-graphs in their recent paper [5]. They defined a q-graph Q by its incidence matrix, an $n \times e$ matrix, where $e = |E(Q)|$ is the number of edges, $n = |V(Q)|$ is the number of vertices, and each column contains exactly two elements from $\{1, \ldots, q\}$, every other element of the matrix being zero. Every column of the incidence matrix is assumed to be distinct. This is a generalization of the incidence matrices of ordinary graphs G, which is exactly the case $q = 1$ with their definitions.

[1]Research is supported by Hungarian REU 2022
[2]Research is supported by Hungarian REU 2022
[3]Research is supported by Hungarian REU 2022
[4]Research is supported by Hungarian REU 2022

For each q-edge $e \in E(Q)$, the *support* of e is the two vertices (i.e. row indices) in its column with non-zero values. The support of e is denoted as S_e. Using this notation, the formal definition of q-graphs can be given in the following way:

Definition 1 (Patkós, Tuza, Vizer) *$\mathcal{Q}(n,r) = \{\boldsymbol{x} \in \{0,1,\dots,q\}^n : |S_{\boldsymbol{x}}| = r\}$. A q-graph H on n vertices is $H \subseteq \mathcal{Q}(n,2)$. The vertex set of H is $\bigcup_{\boldsymbol{x}\in H} S_{\boldsymbol{x}}$. A q-edge of H is $\boldsymbol{x} \in H$. For $1 \le i \le n$, x_i will denote the i-th coordinate of the vector x.*

It is easy to see that q-graphs indeed contain ordinary graphs as the $q = 1$ special case.

For the purpose of generalising the extremal number $\mathrm{ex}(n,F)$ to the case of q-graphs, the authors of [5] first determined when a q-graph Q contains an ordinary graph F. Two q-edges $e, f \in Q$ are said to s-intersect at the vertex v, if $v \in S_e$, $v \in S_f$ and the sum of the entries of the incidence matrix at (e,v) and (f,v) is at least s. The q-graph Q is thus said to contain an s-copy of the ordinary graph F, if there is a set of q-edges in Q which is isomorphic to F, and each pair of incident edges s-intersect. Formally,

Definition 2 (Patkós, Tuza, Vizer) *Let $F = (v(F), E(F))$ be an ordinary graph without isolated vertices, and $H \subseteq \mathcal{Q}(n,2)$ be a q-graph on n vertices. Then H is an* s-copy *of F if $(\bigcup_{\mathbf{x}\in\mathbf{H}} S_{\mathbf{x}}, \{S_{\mathbf{x}} : \mathbf{x} \in \mathbf{H}\})$ is isomorphic [1] to F, and there exists an isomorphism $\iota : F \to (\bigcup_{\mathbf{x}\in\mathbf{H}} S_{\mathbf{x}}, \{S_{\mathbf{x}} : \mathbf{x} \in \mathbf{H}\})$ such that for every $uv, wv \in E(F), u \neq w$, it holds that the q-edges x, x' in H with $S_x = \{\iota(u), \iota(v)\}, S_{x'} = \{\iota(w), \iota(v)\}$ satisfy the condition $x_{\iota(v)} + x'_{\iota(v)} \ge s$.*

If F contains isolated vertices, then $H \subseteq \mathcal{Q}(n,2)$ is said to be an s-copy of F if $n \ge |V(F)|$ and H is an s-copy of $F[U]$, where U is the set of non-isolated vertices in F.

Now we are ready to define the Turán number for q-graphs:

Definition 3 (Patkós, Tuza, Vizer) *For a graph F and integers $n, q, s \ge 1$,*

$$ex(n,F,q,s) = max\{|H| : H \subseteq \mathcal{Q}(n,2),\ H \text{ does not contain an } s\text{-copy of } F\}.$$

Furthermore,

$$EX(n,F,q,s) = \{H : H \subseteq \mathcal{Q}(n,2), H \text{ does not contain an } s\text{-copy of } F,\ |H| = ex(n,F,q,s)\}.$$

Again, we emphasize that this definition includes the extremal number for ordinary graphs: substituting $q = 1, s = 2$ in the above formula yields $\mathrm{ex}(n,F)$.

Very much like in [5], we only addressed the case $s = q+1$, for which we introduce a special notation:

Notation 4 (Patkós, Tuza, Vizer) *$ex(n,F,q,q+1) = ex(n,F,q)$, $EX(n,F,q,q+1) = EX(n,F,q)$*

The reason behind this is that most of the other cases are redundant, or can be retraced to $s = q+1$. Take, for instance, an ordinary graph F without vertices of degree one, and $s = q+2$. Then, in a q-graph in $\mathrm{EX}(n,F,q,q+2)$, we may include every q-edge where at least one of the labels is 1, as these edges cannot be present in a $q+2$-copy of G, since $1 + x_v \le 1 + q < q + 2$ for every q-edge x. Therefore, we only need to pay attention to the remaining q-edges with labels from $\{2,3,\dots,q\}$, which (after identifying $i \in \{2,\dots,q\}$ with $i-1 \in \{1,\dots,q-1\}$) is equivalent to having a $q-1$-graph H' without a q-copy of F; thus reducing the problem to finding $\mathrm{ex}(n,F,q-1,q) = \mathrm{ex}(n,F,q-1)$.

The following definition includes some useful notations that will be used in the upcoming sections.

Definition 5 (Patkós, Tuza, Vizer) *For $H \in \mathcal{Q}(n,2)$, and $(a,b) \in [q]^2$, let $\overrightarrow{H}_{a,b}$ be the directed graph on $[n]$ with edges (i,j) for which the q-edge $\mathbf{x}$ with $S_{\mathbf{x}} = \{i,j\}, x_i = a, x_j = b$ appears in H. For $a,b,c,d \in [q]$, let $\overrightarrow{H}_{(a,b),(c,d)} = \overrightarrow{H}_{a,b} \cap \overrightarrow{H}_{c,d}$. Finally, let $H_{a,b}$ and $H_{(a,b),(c,d)}$ the graphs obtained by first removing orientations, and then the multiple edges from $\overrightarrow{H}_{a,b}$ and $\overrightarrow{H}_{(a,b),(c,d)}$, respectively.*

[1] We call an $\iota : F \to (V,E)$ an isomorphism if $\iota : V(F) \to V$ is a bijection that induces $\iota : E(F) \to E$ such that $\iota(e) = \{\iota(v) : v \in e\} \forall e \in E(F)$

A fundamental part of the proofs rely on a partition of the q-edges with respect to the size of their two labels. Intuitively, one may wish to separately study the "large" q-edges and the others, as it turns out to be easier to handle them that way. The next definition formalizes the notion of these "large" edges of a q-graph.

Definition 6 (Patkós, Tuza, Vizer) *For a q-graph $H \subseteq \mathcal{Q}(n,2)$, let*

$$H^L = \{\boldsymbol{x} \in H : S_{\boldsymbol{x}} = \{i,j\},\, x_i, x_j \geq \frac{q+1}{2}\}.$$

The authors of [5] have established numerous results, with a main focus on the $q = 2$ case. They have given an upper bound of $\mathrm{ex}(n,F,q)$ when F is a tree, and showed that in the $q = 2$ case, a well-known construction in extremal graph theory yields the optimal value. Moreover, as a first step towards the general case $F = C_{2k+1}$, they computed $\mathrm{ex}(n,C_3,2)$.

Here we list a selection of their theorems, which will be used or generalized in the upcoming sections.

Proposition 7 (Patkós, Tuza, Vizer) *For any $n \in \mathbb{N}$ and graph F, we have*

$$\mathrm{ex}(n,F,q) \geq q^2 \cdot \mathrm{ex}(n,F).$$

This first result establishes a trivial lower bound for the Turán number, which hardly ever turns out to be sharp. The case of C_3 is a counterexample, where we use this trivial bound to strengthen the following theorem of Patkós et al..

Theorem 8 (Patkós, Tuza, Vizer) *For $n \geq 2$,*

$$ex(n,C_3,2) = 4 \cdot ex(n,C_3) = 4\left\lfloor \frac{n^2}{4} \right\rfloor.$$

Theorem 9 (Patkós, Tuza, Vizer) *Suppose T is a tree of radius r.*

(1) If the diameter of T is $2r$, then $ex(n,T,2) = (1+o(1)) \cdot \left(\binom{n}{2} + t_{n,r}\right)$.

(2) If the diameter of T is $2r-1$, then $ex(n,T,2) = (1+o(1)) \cdot \left(\binom{n}{2} + t'_{n,r} - \binom{\lfloor \frac{n}{2r-1} \rfloor}{2}\right)$.

Here, $t_{n,r}$ denotes the number of edges in the r-partite Turán graph [2] on n vertices, and $t'_{n,r}$ is the number of edges in the complete r-partite graph on n vertices, where one class has size $\lfloor \frac{n}{2r-1} \rfloor$, and the other class sizes differ by at most one.

Theorem 10 (Patkós, Tuza, Vizer) *For integers $1 \leq r \leq s \leq t$ with $t \geq 2$, we have that:*

(1) If $r = 1$ or $s \leq 2$, then $ex(n,K_{r,s,t},2) = (3+o(1)) \cdot \binom{n}{2}$.

(1) If $r = 2$ or $s \geq 3$, then $ex(n,K_{r,s,t},2) = \left(\frac{13}{4} + o(1)\right) \cdot \binom{n}{2}$.

(1) If $r \geq 3$, then $ex(n,K_{r,s,t},2) = \left(\frac{7}{2} + o(1)\right) \cdot \binom{n}{2}$.

Theorem 11 (Patkós, Tuza, Vizer) *Suppose F is a bipartite pseudo-forest [3], and at least one of its connected components contains a cycle. Then*

$$ex(n,F,q) = \left(\left\lfloor \frac{q^2}{2} \right\rfloor + o(1)\right)\binom{n}{2}.$$

[2] The Turán graph $T_{n,r}$ is a complete r-partite graph on n vertices with $\lfloor \frac{n}{r} \rfloor$ and $\lceil \frac{n}{r} \rceil$ class sizes.

[3] A pseudo-forest is a graph for which every connected component is comprised of either a tree, or a tree plus an edge.

Our main contribution to the topic simply states that for every graph F and even q, it suffices to examine the $q = 2$ case. This has a long-reaching impact, as combining it with the results of Patkós et al. significantly narrows down the unknown values of $\mathrm{ex}(n, F, q)$, at least when q is even.

Theorem 12 *For every even q and ordinary graph F, $ex(n, F, q) = \frac{q^2}{4} \cdot \mathrm{ex}(n, F, 2)$.*

The proof is comprised of a somewhat technical part where we explain how **Lemma** 23 provides an optimal q-graph with a special structure; and a part where we exploit that structure to connect the general setup to the $q = 2$ case.

The upcoming statements make use of **Theorem 12**, and transcribe the above listed theorems in [5] from $q = 2$ to even values of q.

Theorem 13 *Suppose T is a tree of radius r, and q is even.*

(1) If the diameter of T is $2r$, then $ex(n, T, q) = \left(\frac{q^2}{4} + o(1)\right) \cdot \left(\binom{n}{2} + t_{n,r}\right)$.

(2) If the diameter of T is $2r-1$, then $ex(n, T, q) = \left(\frac{q^2}{4} + o(1)\right) \cdot \left(\binom{n}{2} + t'_{n,r} - \binom{\lfloor \frac{n}{2r-1} \rfloor}{2}\right)$.

Proposition 14 *For integers $1 \leq r \leq s \leq t$ with $t \geq 2$, and an even q, we have that:*

(1) If $r = 1$ or $s \leq 2$, then $ex(n, K_{r,s,t}, q) = \left(\frac{3q^2}{4} + o(1)\right) \cdot \binom{n}{2}$.

(1) If $r = 2$ or $s \geq 3$, then $ex(n, K_{r,s,t}, q) = \left(\frac{13q^2}{16} + o(1)\right) \cdot \binom{n}{2}$.

(1) If $r \geq 3$, then $ex(n, K_{r,s,t}, q) = \left(\frac{7q^2}{8} + o(1)\right) \cdot \binom{n}{2}$.

The proof of **Theorem 12** strongly relies on the parity of q, so in the general case, when q is not necessarily even, the same reasoning will not suffice. For now, we must settle for an upper bound when q is odd:

Proposition 15 $ex(n, F, q) \leq \frac{q^2}{4} \cdot ex(n, F, 2), \ \forall q \in \mathbb{N}$

By itself, **Proposition** 15 does not carry a huge significance, as it only provides an upper bound for $\mathrm{ex}(n, F, q)$, but in some special cases it coincides with the trivial lower bound $q^2 \cdot \mathrm{ex}(n, F)$, hence giving the exact value of $\mathrm{ex}(n, F, q)$; as is the case with C_3. We present a generalization of **Theorem 8**.

Proposition 16 $\mathrm{ex}(n, C_3, q) = q^2 \cdot \lfloor \frac{n^2}{4} \rfloor, \ \forall q \in \mathbb{N}$.

The statement easily follows from the combination of two previous propositions: On one hand, we know from **Proposition** 7 that $\mathrm{ex}(n, C_3, q) \geq q^2 \cdot \mathrm{ex}(n, C_3) = q^2 \lfloor \frac{n^2}{4} \rfloor$; and on the other hand, $\mathrm{ex}(n, C_3, q) \leq \frac{q^2}{4} \cdot \mathrm{ex}(n, C_3, 2) = q^2 \cdot \lfloor \frac{n^2}{4} \rfloor$ comes from **Proposition 15**.

As suggested by Patkós, Tuza and Vizer in [5], the next step in our research was to determine $\mathrm{ex}(n, C_{2k+1}, 2)$ at least asymptotically.

Proposition 17 $\mathrm{ex}(n, C_{2k+1}, 2) = \left(\lfloor \frac{2^2}{2} \rfloor + o(1)\right) \cdot \binom{n}{2} = n^2 + o(n^2)$.

The core of the proof stems from the same idea as in **Proposition 15**; namely, to use the already established results for $2(2k+1)$ (or in this case, $2(2k-1)$).

Now that we have the asymptotic value of $\mathrm{ex}(n, C_{2k+1}, 2)$, we can assert the conjecture of Patkós et al. that $\mathrm{ex}(n, C_{2k+1}, q)$ is asymptotically $q^2 \cdot \mathrm{ex}(n, C_{2k+1})$.

Proposition 18 *For every $q \geq 2$, $ex(n, C_{2k+1}, q) = \frac{n^2}{4} \cdot q^2 + o(n^2)$.*

PROOF: The proof is the same as for C_3; we only need to compare the upper bound provided by **Proposition 15** with the trivial lower bound in **Proposition 7**. It follows that

$$q^2 \cdot \left\lfloor \frac{n^2}{4} \right\rfloor = q^2 \cdot \mathrm{ex}(n, C_{2k+1}) \leq \mathrm{ex}(n, C_{2k+1}, q) \leq \frac{q^2}{4} \cdot (n^2 + o(n^2)) = q^2 \cdot \frac{n^2}{4} + o(n^2).$$

□

Finally, let us combine **Proposition** 15 with the monotonicity of $\mathrm{ex}(n, F, q)$ in q:

Proposition 19 *For every graph F and $q \geq 2$,*

$$\frac{(q-1)^2}{4} \cdot ex(n, F, 2) \leq ex(n, F, q) \leq \frac{q^2}{4} \cdot ex(n, F, 2)$$

PROOF: The second inequality is simply **Proposition** 15. If q is even, then $\mathrm{ex}(n, F, q)$ equals to the right hand side, and if q is odd, then $q-1$ is even, so **Proposition** 12 is applicable: $\frac{(q-1)^2}{4} \cdot \mathrm{ex}(n, F, 2) = \mathrm{ex}(n, F, q-1) \leq \mathrm{ex}(n, F, q)$. □

This limits $\mathrm{ex}(n, F, q)$ to an interval of size $\frac{2q-1}{4} \cdot \mathrm{ex}(n, F, 2)$. When q is odd, we feel that this boundary can be improved both ways, as the proof does not take into account the specific attributes of the q-graph. An equality in either side would entail that an optimal construction for $q-1$ or $2q$ is simultaneously the best one can achieve for q. We suspect this is not the case, and there is some room for improvement.

2 Proofs

Our first observation establishes a monotonic property of $\mathrm{ex}(n, F, q)$.

Proposition 20 *For $m \leq n$,*

$$\frac{\mathrm{ex}(n, F, q)}{n(n-1)} \leq \frac{\mathrm{ex}(m, F, q)}{m(m-1)}.$$

PROOF: Let $H \in \mathrm{EX}(n, F, q)$ and count the number of pairs (V', e) where $V' \subset V(H)$ is of size m, $e \in E(H)$ and e has support in V'. Note that each edge in H gets counted $\binom{n-2}{m-2}$ times and for any V' there are at most $\mathrm{ex}(m, F, q)$ many q-edges in H with support in V'. The statement follows. □ Since $\lfloor \frac{n^2}{4} \rfloor \sim \frac{n(n-1)}{4}$, we obtain the following corollary.

Corollary 21 *For any graph F and $q \geq 1$ the limit $\lim_{n\to\infty} \frac{\mathrm{ex}(n,F,q)}{\lfloor n^2/4 \rfloor}$ exists.*

The key feature of our paper is the reduction of the case when q is even to the $q = 2$ case. From here, a simple reasoning (using the aforementioned reduction) gives that $\frac{q^2}{4} \cdot \mathrm{ex}(n, F, 2)$ is always an upper bound for arbitrary values of q. Although we expect that this upper bound is far from being sharp in a general setup, it comes in handy for $F = C_3$, as we will see in the proof for **Proposition** 16.

The fundamental part of our proof is establishing a connection between the problem of determining $\mathrm{ex}(n, F, q)$ and a problem for ordinary graphs which is closely related to the fractional vertex covering problem. In fact, the traditional way to show that there always exists a half-integral minimal vertex cover can be applied to our case with little to no change. However, we present another approach to prove the next statement.

Lemma 22 *Let G be an ordinary graph. Then there is a function y taking values in $\{0, \lfloor \frac{q}{2} \rfloor, \lceil \frac{q}{2} \rceil, q\}$ maximizing $\sum_{u \in V(G)} x(u)$ on the set $L(G) = \left\{ x : V(G) \to \{0\} \cup [q] \quad | \quad x(u) + x(v) \leq q \ \forall uv \in E(G) \right\}$.*

PROOF: At first we assume that there is a set of independent vertices $A \subset V(G)$ so that $|A| > |N(A)|$. We may pick a minimal such A, that is, $|B| \leq |N(B)|$ $\forall B \subsetneq A$. By assumption A is nonempty. Let $B = A \setminus \{v\}$ for some arbitrary $v \in A$. Then $|A|-1 = |B| \leq |N(B)| \leq |N(A)| < |A|$ so $N(B) = N(A)$. By Hall's theorem there is a matching $M \subset E(G)$ from B to $N(A)$. For $x \in L(G)$ we get $\sum_{u \in A \cup N(A)} x(u) = x(v) + \sum_{u \in B \cup N(B)} x(u) \leq q + |B| \cdot q = q|A|$, since $x(u) + x(v) \leq q$ for $uv \in M$. Since A is minimal, the bipartite graph induced by G on $A \sqcup N(A)$ is connected. So equality holds above if and only if $x|_A \equiv q$ and $x|_{N(A)} \equiv 0$. Since there are no edges from A to $V(G) \setminus (A \cup N(A))$, any maximal $x \in L(G)$ is the union of $(A \times \{q\}) \cup (N(A) \times \{0\})$ and some maximal $x' \in L(G')$ where $G' = G \setminus (A \cup N(A))$. By repeating the argument if G' has an independent $A' \subset V(G')$ with $|A'| > |N(A')|$, any maximal $x \in L(G)$ is the union of $(V_1 \times \{q\}) \cup (V_2 \times \{0\})$ and some maximal $x' \in L(G_1)$, where V_1 and V_2 are the disjoint subsets of $V(G)$ that we obtain by the argument, $G_1 = G \setminus (V_1 \cup V_2)$ and $|H| \leq |N(H)|$ for all independent $H \subset V(G_1)$. Observe that $S = \{u \in V(G_1) \mid x(u) > \lfloor \frac{q}{2} \rfloor\}$ is an independent set of vertices in G_1 for $x \in L(G_1)$. By Hall's theorem there is a matching from S to some $T \subset N(S)$ in G_1, so we calculate

$$\sum_{u \in V(G_1)} x(u) \;\leq \sum_{u \in S \cup T} x(u) \;+ \sum_{u \notin S \cup T} x(u) \leq q|S| + \left\lfloor \frac{q}{2} \right\rfloor \big(|V(G_1)| - 2|S|\big).$$

Note that this maximum is achieved by $x' \equiv \frac{q}{2}$ if q is even, and

$$x'(u) = \begin{cases} \lceil \frac{q}{2} \rceil & \text{if } u \in S \\ \lfloor \frac{q}{2} \rfloor & \text{if } u \notin S, \end{cases}$$

and this x' is in $L(G_1)$. So there is a maximal $x' \in L(G_1)$ taking values in $\big\{0, \lfloor \frac{q}{2} \rfloor, \lceil \frac{q}{2} \rceil, q\big\}$. The statement follows. □

We now generalize **Lemma** 22 to hypergraphs. There are many ways to do this, the one we discuss here is the case that is useful for us in the setting of q-graphs.

Lemma 23 *Let $\mathcal{H} = (V, H)$ be a hypergraph, and let $x : V \to \{0\} \cup [q]$ a function on the vertices of $\mathcal{H}$ such that it satisfies the following condition: $\forall h \in H$ $\exists u, v \in V(h) : x(u) + x(v) \leq q$. Then an x that maximizes the expression $\mathbf{1} \cdot \mathbf{x}$ can be chosen to have values from the set $\{0, \lfloor \frac{q}{2} \rfloor, \lceil \frac{q}{2} \rceil, q\}$.*

PROOF: Construct an ordinary graph G on the vertex set $V(\mathcal{H})$ as follows: choose a pair of vertices $\{u, v\}$ from each hyperedge $h \in H$ (this pair will guarantee the sum condition of x for h), and add the edge (u, v) to G. By applying **Lemma** 22, we can set $x_G = \arg\min\{\mathbf{1} \cdot x : x(u) + x(v) \leq q, \forall uv \in E(G)\}$ to have values from $\{0, \lfloor \frac{q}{2} \rfloor, \lceil \frac{q}{2} \rceil, q\}$. It is easy to see that if we take the solution x for which $\mathbf{1} \cdot x = \min_G\{\mathbf{1} \cdot x_G\}$ over every possible choice of G, we get an optimal solution for the original problem for $\mathcal{H}$. □

The main result of this paper, **Theorem 12**, simply states that for every graph F and even q, it suffices to examine the case $q = 2$. Armed with the previous lemma, we are ready to prove the theorem.

PROOF:[Proof of Theorem 12]

For the sake of simplicity, let us use a temporary notation for q-edges. Let $(u, v, a, b) \in \mathcal{Q}(n, 2)$ be the q-edge x where $S_x = \{u, v\}$ and $x_u = a, x_v = b$.

Consider a $H \subseteq \mathcal{Q}(n, 2)$ without a $(q+1)$-copy of F with $\mathrm{ex}(n, F, q)$ q-edges. Let v be an arbitrary vertex of H. For another vertex $u \neq v$ and $i \in [q]$, let $m(u, i) = \max_{r \in [q]}\{(u, v, i, r) \in H\}$. We intend to alter the q-edges adjacent to v in a way that every $m(u, i)$ will become $m'(u, i) \in \{0, \frac{q}{2}, q\}$; and in the meantime, the total number of q-edges does not decrease. For that purpose, let $x_{(u,i)}$ be a variable reflecting the current value of $m(u, i)$. The condition that H is $(q+1)$-F-free implies restrictions for certain variables: if the set of q-edges

$$L = \{(u_k, v, r_k, x_{(u_k, r_k)}) |\, u_k \in S_H, r_k \in [q], k = 1, 2, \ldots\}$$

could be part of a potential $(q+1)$-copy of F, then at least one of the following inequalities must hold: $\{x_{(u_i,r_i)}+x_{(u_j,r_j)} \leq q,\ (u_i,v,r_i,x_{(u_i,r_i)}),(u_j,v,r_j,x_{(u_j,r_j)}) \in L, i \neq j\}$. Let $\mathcal{S}$ denote the union of variable sets $\{x_{(u_1,r_1)}, x_{(u_2,r_2)},\ldots\}$ for which we have a restriction in the above form.

In the following part of the proof, we construct a hypergraph $\mathcal{H}$, in which one can encode the properties of the q-edges with common endpoint v. Let $V(\mathcal{H}) = \{w_{u,i} | u \in N_H(v), i \in [q]\}$, and $E(\mathcal{H}) = \{(w_{u_1,r_1}, w_{u_2,r_2},\ldots,w_{u_j,r_j}) : \{x_{(u_1,r_1)},\ldots,x_{(u_k,r_k)}\} \in \mathcal{S}\}$. The hypergraph $\mathcal{H}$ and the function x satisfy the conditions of **Lemma** 23; so, bearing in mind that now $\lfloor \frac{q}{2} \rfloor = \lceil \frac{q}{2} \rceil = \frac{q}{2}$, we can change the entries of x to $0, \frac{q}{2}$ and q. In the meantime, we can alter the q-edges adjacent to v according to the change in x so that $\max_{r\in[q]}\{(u,v,i,r) \in H\}$ becomes $m'(u,i) \in \{0, \frac{q}{2}, q\}$ for every $u \in N_H(v)$ and $i \in [q]$. Meanwhile, the number of q-edges attached to v does not decrease:

$$\sum_{u\in N_H(v)} \sum_{i\in[q]} m(u,i) = \mathbf{1}\cdot x \leq \max_x \mathbf{1}\cdot x = \sum_{u\in N_H(v)} \sum_{i\in[q]} m'(u,i).$$

By iterating the above modification for every vertex v in $[n]$, we end up with a q-graph H' that has the following property:

$$\forall (u,v,a,b) \in H' : \max_{r\in[q]}\{(u,v,r,b) \in H'\} \in \left\{0, \frac{q}{2}, q\right\}, \max_{r\in[q]}\{(u,v,a,r) \in H'\} \in \left\{0, \frac{q}{2}, q\right\}.$$

Indeed, suppose that we have already processed the q-edges adjacent to v, meaning that for the current q-graph H, it holds that for every $u \in N_H(v)$ and $a \in [q] : \max_{r\in[q]}\{(u,v,a,r)\} \in \{0, \frac{q}{2}, q\}$. When we arrive at processing the node u, we may replace the label a of a q-edge (u,v,a,b), but the label b at node v remains the same. This implies that $\max_{r\in[q]}\{(u,v,a,r)\} \in \{0, \frac{q}{2}, q\}$ remains true for every $u \in N_H(v)$ and $a \in [q]$.

Note that the modified H' is still optimal, so if a q-edge (u,v,a,b) is in H', then so is every other (u,v,a',b') with $a' \leq a$, $b' \leq b$. With this remark, the special structure of H' can be rephrased in the following way: for every support $\{u,v\}$, consider the following partition of potential q-edges:

- $E_{s,s} = \{(u,v,a,b) : 1 \leq a \leq \frac{q}{2}, 1 \leq b \leq \frac{q}{2}\}$, $|E_{s,s}| = \frac{q^2}{4}$
- $E_{b,s} = \{(u,v,a,b) : \frac{q}{2} < a \leq q, 1 \leq b \leq \frac{q}{2}\}$, $|E_{b,s}| = \frac{q^2}{4}$
- $E_{s,b} = \{(u,v,a,b) : 1 \leq a \leq \frac{q}{2}, \frac{q}{2} < b \leq q\}$, $|E_{s,b}| = \frac{q^2}{4}$
- $E_{b,b} = \{(u,v,a,b) : \frac{q}{2} < a \leq q, \frac{q}{2} < b \leq q\}$, $|E_{b,b}| = \frac{q^2}{4}$.

We may observe that if there is a q-edge (u,v,a,b) from $E_{s,s}$ in H', then $E_{s,s} \subseteq H'$ must hold; and a similar statement is true for $E_{s,b}, E_{b,s}$ and $E_{b,b}$. For each support pair $\{u,v\}$, let us identify these four q-edge sets with the $(u,v,1,1), (u,v,2,1), (u,v,1,2)$ and $(u,v,2,2)$ 2-edges of a 2-graph H'', and denote the number of these 2-edges in H'' by $e_{s,s}, e_{b,s}, e_{s,b}, e_{b,b}$ respectively. Then

$$\mathrm{ex}(n,F,q) = |H'| = |E_{s,s}|\cdot e_{s,s} + |E_{s,b}|\cdot e_{s,b} + |E_{b,s}|\cdot e_{b,s} + |E_{b,b}|\cdot e_{b,b} = \frac{q^2}{4}\cdot |H''|.$$

H'' cannot contain a 3-copy of F, because H' did not contain a $(q+1)$-copy of F, so $|H''| \leq \mathrm{ex}(n,F,2)$, and

$$\mathrm{ex}(n,F,q) \leq \frac{q^2}{4}\cdot \mathrm{ex}(n,F,2).$$

For the other direction, consider a 2-graph $H \in \mathrm{EX}(n,F,2)$. One only needs to reverse the above construction: substituting the edges $(u,v,1,1), (u,v,1,2), (u,v,2,1), (u,v,2,2)$ in H with the edge sets $E_{s,s}, E_{s,b}, E_{b,s}, E_{b,b}$ respectively gives a q-graph H' with $|H'| = \frac{q^2}{4}\cdot|H| = \frac{q^2}{4}\cdot \mathrm{ex}(n,F,2)$. H' does not contain a $(q+1)$-copy of F, so $\mathrm{ex}(n,F,q) \geq |H'| = \frac{q^2}{4}\cdot \mathrm{ex}(n,F,2)$.

□

For odd values of q, there is no easy way to interpret a mapping of $\{0, \lfloor \frac{q}{2} \rfloor, \lceil \frac{q}{2} \rceil, q\}$ to the values $\{0, 1, 2\}$ the same way as in the previous reasoning. The thorough examination of the q-edges $\{\left(\lceil \frac{q}{2} \rceil, a\right) : a \in [q]\}$ and $\{\left(\lfloor \frac{q}{2} \rfloor, a\right) : a \in [q]\}$ might provide better answers than **Proposition 15**, as the proof consists of a simple reduction from q to $2q$, and does not use the underlying structure of the q-graph.

PROOF:[Proof of Proposition 15] Consider a q-graph $H \in \mathrm{EX}(n, F, q)$ and define H' as

$$H' = \{(u, v, 2a, 2b), (u, v, 2a-1, 2b), (u, v, 2a, 2b-1), (u, v, 2a-1, 2b-1) : (u, v, a, b) \in H\}.$$

In the obtained $2q$-graph H', a $(2q+1)$-copy of F does not appear: the largest value of s for which an s-copy of F is present in H is at most q, so the largest value of s for which an s-copy of F is present in H' is at most $2q$. Hence, by **Theorem 12**,

$$|H'| = 4 \cdot |H| = 4 \cdot \mathrm{ex}(n, F, q) \leq \mathrm{ex}(n, F, 2q) = q^2 \cdot \mathrm{ex}(n, F, 2).$$

□

For the final part of this section, we prove **Proposition 17**. Again, let (u, v, a, b) denote the q-edge x with $S_x = \{u, v\}$ and $x_u = a, x_v = b$. Let us recall from **Definition 5** that $\overrightarrow{H}_{a,b} = \{(u, v) \in [n]^2 : (u, v, a, b) \in H\}$ with an edge (u, v) directed from u to v, $H_{a,b} = \{(u, v) \in [n]^2 : (u, v, a, b) \in H \text{ or } (v, u, a, b) \in H\}$, and $H_{(a,b),(c,d)}$ is $\overrightarrow{H}_{a,b} \cap \overrightarrow{H}_{c,d}$ without orientations and multiple edges. We will use a result of Zhou and Li, namely, **Theorem 1.8.** from [4], and **Lemma 3.1.** from [5].

Theorem 24 (Li, Zhou) *Let $k, n \in \mathbb{N}^*, n = qk + r, 0 \leq r < k$, and let $\overrightarrow{C_{k+1}}$ be the directed cycle on $k+1$ vertices. Then*

$$ex(n, \overrightarrow{C_{k+1}}) = \frac{1}{2}n^2 + \frac{k-2}{2}n - \frac{r(k-r)}{2}.$$

Lemma 25 (Patkós, Tuza, Vizer) *Let G be a bipartite graph such that all its components are unicyclic or trees. Suppose $H \subseteq \mathcal{Q}(n, 2)$ does not contain any $(q+1)$-copy of graph G. Then for any $(a, b) \in [q]^2$, the graph $H_{(a,b),(q+1-a,q+1-b)}$ has $o(n^2)$ edges.*

PROOF:[Proof of Proposition 17] Let H be an optimal 2-graph without a 3-copy of C_{2k+1}. To bound the number of 2-edges in $\overrightarrow{H}_{1,2}$, we rely on **Theorem 24**. It implies that

$$|\overrightarrow{H}_{1,2}| = \frac{1}{2}n^2 + \frac{2k-1}{2}n + O(1) = \frac{1}{2}n^2 + o(n^2).$$

Now we turn to examine $H_{1,1} \cup H_{2,2}$, where an edge is included twice if it is both in $H_{1,1}$ and $H_{2,2}$. We prove by induction on n that $|H_{1,1} \cup H_{2,2}| = \frac{1}{2}n^2 + o(n^2)$. Suppose for contradiction that $H_{1,1} \cup H_{2,2}$ has more edges. Then substituting $q = 2, a = b = 1$ into **Lemma 25** gives a 3-copy of a C_{4k-2}; otherwise $|H_{1,1} \cap H_{2,2}| = o(n^2)$ would hold, and since the optimal property of H entails $H_{1,1} \cap H_{2,2} = H_{2,2}$, $|H_{1,1} \cup H_{2,2}| = |H_{1,1}| + |H_{2,2}| = |H_{1,1}| + |H_{1,1} \cap H_{2,2}| \leq \binom{n}{2} + o(n^2)$ would stand. Consequently, a 3-copy of C_{4k-2} is indeed present in $H_{1,1} \cup H_{2,2}$.

Let us denote the support of this 3-copy by S_{4k-2}, and the vertices in S_{4k-2} by $v_1, v_2, \ldots, v_{4k-2}$. As the pair of 2-edges attached to v_1 in the cycle 3-intersect, at least one of them must have labels $(2, 2)$. The same stands for the 2-edges attached to v_{2k}. Using the symmetry of C_{4k-2}, this can happen in one of two ways: either $(v_1, v_2, 2, 2)$ and $(v_{2k}, v_{2k-1}, 2, 2)$ are in H, or $(v_1, v_2, 2, 2)$ and $(v_{2k}, v_{2k+1}, 2, 2)$. Consider an arbitrary vertex v in $S_H \setminus S_{4k-2}$. If the 2-edges $(v_1, v_2, 2, 2)$ and $(v_{2k}, v_{2k-1}, 2, 2)$ are in H, then we need to omit at least one 2-edge from both
$\{(v, v_1, 2, 2), (v, v_{2k}, 1, 1)\}$ and $\{(v, v_1, 1, 1), (v, v_{2k}, 2, 2)\}$, or else a 3-copy of C_{2k+1} would be formed by $vv_1v_2 \ldots v_{2k}v$. Otherwise, the 2-edges $(v_1, v_2, 2, 2)$ and $(v_{2k}, v_{2k+1}, 2, 2)$ are in the 3-copy of C_{4k-2}. Then H can only contain at most one q-edge from both of the pairs $\{(v, v_1, 2, 2), (v, v_{2k}, 1, 1)\}$ (or else a 3-copy of C_{2k+1} would be formed by $vv_{2k}v_{2k+1} \ldots v_{4k-2}v$) and $\{(v, v_1, 1, 1), (v, v_{2k}, 2, 2)\}$ (or else a 3-copy of

C_{2k+1} would be formed by $vv_1v_2\dots v_{2k}v$). In both cases, we may conclude that out of the 4 possible 2-edges in $H_{1,1} \cup H_{2,2}$ between v and $\{v_{2k}, v_1\}$, at most 2 may be included in H.

The same reasoning can be repeated for v_i and $v_{i+2k-1(\mathrm{mod}\ 4k-2)}$ instead of v_1 and v_{2k}. By summing it up for $\{v_i, v_{i+2k-1(\mathrm{mod}\ 4k-2)} : i = 1, 2, \dots, 2k-1\}$, and for every $v \in S_H - S_{4k-2}$, we gain that at most half of the potential 2-edges between S_{4k-2} and $S_H - S_{4k-2}$ can be present in $H_{1,1} \cup H_{2,2}$. Applying the induction hypothesis shows that $\frac{1}{2}|S_H - S_{4k-2}|^2 + o(n^2)$ 2-edges can be spanned by $S_H - S_{4k-2}$ in $S_{H_{1,1}} \cup S_{H_{2,2}}$. The number of 2-edges spanned by H in S_{4k-2} can be bounded by a constant which is independent from n. Consequently, the total number of 2-edges in $H_{1,1} \cup H_{2,2}$ amounts to

$$2 \cdot |S_{4k-2}| \cdot |S_H - S_{4k-2}| + \frac{1}{2}|S_H - S_{4k-2}|^2 + o(n^2) =$$

$$= \frac{1}{2}|S_H \cup S_{4k-2}|^2 + o(n^2) = \frac{1}{2}n^2 + o(n^2),$$

which concludes our inductive proof. Finally, $|H| = |H_{(1,1)} \cup H_{(2,2)}| + |\overrightarrow{H}_{1,2}| = n^2 + o(n^2)$.

□

3 Concluding remarks and questions

We conclude this paper by highlighting some questions that can be articulated in relation to q-graphs. As a consequence of our results, one may immediately transfer every statement for $q = 2$ to every even q. Moreover, an upper and a lower bound is established for the general case as well. However, we need to emphasize that these constraints may not provide a precise value of the Turán number for every graph F, as we suspect is the case for trees. Consequently, the exact (or asymptotic) value of $\mathrm{ex}(n, F, q)$ remains unknown for odd values of q. A potential next step could be to improve our trivial bounds, or prove that one of the bounds coincides with $\mathrm{ex}(n, F, q)$.

On a general note, let us highlight that our main achievement was the reduction of even q values. Apart from providing an asymptotical answer for $\mathrm{ex}(n, C_{2k+1}, 2)$, we did not contribute to solving any other questions imposed by Patkós, Tuza and Vizer. Their conjectures and questions remain open, the seemingly most attainable among them being the case of forests.

Finally, let us introduce some alternative definitions that may lead to other interesting topics in relation to q-graphs and Turán numbers. Our way of defining a q-graph allows q-edges with pairs of labels, where the labels can arbitrarily be chosen from the set $\{1, 2, \dots, q\}$. Applying additional constraints when selecting them is a natural first thought while looking for inspiration to come up with new ideas. The q-edges x with $S_x = \{u, v\}$, $x_u + x_v = q+1$ played an important role in some proofs, and in the definition of the universal q-tree (see [5]). One might opt to define a q-graph as a collection of q-edges whose labels sum up to $q+1$. Another viable option could be to only allow edges where the two labels are equal, i. e. $x \in \mathcal{Q}(n, 2)$, $S_x = \{u, v\}$, $x_u = x_v$.

Additionally, the inclusion of a $(q+1)$-copy of a graph H in a q-graph can be approached differently. We required that among the q-edges adjacent to a node v, each pair x, y should $(q+1)$-intersect, i. e. $x_v + y_v \geq q+1$ must be true. An alternative version of this inequality is the following: if $x_1, x_2, \dots, x_k$ are all the $q-$edges with endpoint v, then $\sum_{i=1}^{k}(x_k)_v \geq q+1$.

We hope that these alternative definitions can successfully be exploited to gain new results and discover interesting topics in the future.

References

[1] Erdős, P., Simonovits, M., A limit theorem in graph theory, *Studia Sci. Math. Hungar.* **1** (1966), 51-57.

[2] Erdős, P., Stone, A.H., On the structure of linear graphs, *Bulletin of the American Mathematical Society* **52** (1946), 1087-1091.

[3] Füredi, Z., Simonovits, M., The history of degenerate (bipartite) extremal graph problems, *Bolyai Soc. Math. Stud.* **25** (2013), 169-224

[4] Li, B., Zhou, W., The Turán problems of directed paths and cycles in digraphs, (2021), arXiv:2102.10529v1 [math.CO].

[5] Patkós, B., Tuza, Zs., Vizer, M., Extremal graph theoretical questions for q-ary vectors, Unpublished manuscript [4] (2022)

[6] Turán, P., On an extremal problem in graph theory, *Matematikai és Fizikai Lapok (in Hungarian)* **48** (1941), 436-452.

[7] Turán, P. On the theory of graphs, *Colloq. Math.* **3** (1954), 19-30.

[4] For a short abstract, see *https://conferences.matheo.si/event/37/attachments/164/345/mgtc2022_list_of_abstracts.pdf*

Compiling Packet Programs to dRMT Switches: Theory and Algorithms

Balázs Vass

Budapest University of Technology and Economics and ELKH-BME Information Systems Research Group
vb[at]tmit.bme.hu

Ádám Fraknói

ELTE Eötvös Loránd University
fraknoiadam[at]student.elte.hu

Erika Bérczi-Kovács

Alfréd Rényi Institute of Mathematics, ELTE Eötvös Loránd University, ELKH-ELTE Egerváry Research Group on Combinatorial Optimization
erika.berczi-kovacs[at]ttk.elte.hu

Gábor Rétvári

Budapest University of Technology and Economics and Ericsson Research
retvari[at]tmit.bme.hu

Abstract: This paper considers scheduling-related problems with periodic conditions. The problem is motivated by challenges in P4 program embedding tasks. P4 is a programming language for network devices that decribes how data plane devices such as switches or routers process packets. A critical step in P4 compilation is finding an efficient mapping of the high-level P4 source code constructs to the physical resources exposed by the underlying hardware, while meeting data and control flow dependencies in the program. In this paper, we take a new look at the algorithmic aspects of this problem, with the motivation to understand the fundamental theoretical limits and obtain better P4 pipeline embeddings in the dRMT (disaggregated Match-Action Table) switch architecture. We report mixed results. We find that optimizing P4 program embedding for maximizing throughput is computationally intractable even when some architectural constraints are relaxed, and there is no hope for a tractable approximation with arbitrary precision unless $\mathscr{P}=\mathscr{NP}$. At the same time, we find that the maximal throughput embedding is approximable in polynomial time with a small constant bound. Our evaluations show that the proposed algorithm outperforms the heuristics of prior work both in terms of throughput and compilation speed. [3]

Keywords: reconfigurable switches, algorithmic complexity, approximation algorithms

1 Introduction

P4 is a programming language for network devices which decribes how data plane devices such as switches process packets. dRMT is a programmable switch architecture, designed to handle program execution effectively [2]. At the moment, there are many open algorithmic questions related to P4 program embedding over the dRMT architecture. This is becoming increasingly troubling, since P4 compilation times can easily grow beyond practical. *We initiate the study of the algorithmic landscape of the disaggregated P4 pipeline embedding problem (DPEP)*, where the aim is to find a valid P4 program embedding that maximizes the throughput, or equivalently, minimizes P. To the best of our knowledge, ours is the first principled approach to this end.

We build a sequence of increasingly complex models to characterize the resource requirement for embedding P4 programs into the dRMT pipeline. For each model, we analyze the computational

complexity of the particular incarnation of the P4 pipeline embedding problem, and, using classical results in combinatorial optimization, we derive the corresponding inapproximability (bad news) and approximability (good news) bounds.

Our evaluations show that one of our P4 embedding algorithms, Alg. 1, achieves at least 85% of the theoretically optimal throughput on all P4 programs studied in [2], significantly improving on the heuristic `rnd_sieve` of [2] that achieves only 73% of the theoretical optimum.

2 Models and results

In all of our models, the P4 program is modeled by an Operation Dependency Graph (ODG, [2]) $D = (V, E)$, $V = V_a \cup V_m$, where D is a directed acyclic graph and disjoint set of vertices V_a and V_m represent the match and action nodes, respectively, and arc set E encodes the inter-dependency between the vertices. If the tail of an arc $e = (u, v)$ is a match or action node, then the execution of v can start at least ΔM or ΔA cycles after the start of execution of u, respectively. Moreover, in each CPU cycle, each processor can initiate up to $\overline{M}$ parallel table searches, and can modify up to $\overline{A}$ action fields in parallel. Parameters ΔM, ΔA, $\overline{M}$ and $\overline{A}$ are positive integers. For example, the setting on Fig. 1 can be described by $\Delta M = \Delta A = \overline{M} = 1$, and an arbitrary $\overline{A} \geq 2$.

Also, in line with [2], we restrict our study to cyclic dRMT schedules, where a single packet processing plan is repeated on all packets processed by all processors (cf. [2, Sec. 3.2.]). To give an intuition behind our positive (approximability) results, we anticipate that, based on [2, Theorem 3.5], the dRMT scheduling problem can be simplified to the problem of scheduling a single packet on a single processor. This single packet scheduling has to fulfill a requirement of *P-periodicity*: the set of nodes assigned to clock cycles t, $t + P$, $t + 2P$, ... must meet the ΔM, ΔA, $\overline{M}$, $\overline{A}$ (and later on the width and inter-packet concurrency) requirements together, for all $t \in \{1, \ldots, P\}$.

2.1 BASIC: A simplified model

In the BASIC model, there are no additional constraints to those described above. Every table has a unit width. It is clear that the minimal value of P is at least the maximum of $\lceil |V_m|/\overline{M} \rceil$ and $\lceil |V_a|/\overline{A} \rceil$. As it turns out, the maximum of these two values is reachable with a simple greedy algorithm in $O(|V|log|V| + |E|)$ time (see Theorem 2).

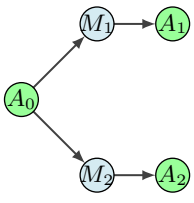
2.2 IPC1: Inter-packet concurrency

On top of the constraints of BASIC, in the IPC1 model, we assume that each processor may start a match for at most a fixed number (*Inter-Packet Concurrency*, IPC) of different packets and likewise start actions for up to IPC different packets. The set of packets that start matches and the set of packets that start actions need not be equal. Below we assume IPC=1. It turns out that in the presence of the IPC constraint, the problem becomes not only $\mathscr{NP}$-hard, but there is also no polynomial time approximation scheme for P (unless $\mathscr{P}$=$\mathscr{NP}$). The $\mathscr{NP}$-hardness and inapproximability can be attained from a well-known $\mathscr{NP}$-hard scheduling problem. On the bright side, in this setting, there exists a 3-approximation algorithm.

Results for this model. *DPEP under the IPC1 model is $\mathscr{NP}$-hard. Bad news: the optimal number of cores to achieve line rate cannot be approximated better than 4/3 unless $\mathscr{P}$=$\mathscr{NP}$. Good news: the optimum can be 3-approximated in polynomial time (see Theorems 3 and 12).*

2.3 WIDTH: Variable table widths

Next, on top of BASIC we will also allow each match and action node to be of arbitrary width, measured by a positive integer $W : V \to \mathbb{N}^+$. We represent this in our WIDTH model by letting each processor to initiate up to $\overline{M}$ parallel table searches in each cycle. It turns out that introducing variable table widths on top of the BASIC model also makes the DPEP $\mathscr{NP}$-hard but constant-factor approximable:



(a)

cycle / proc.	0	1	2	3	4	5	6
0	A_0	M_1	M_2	$A_1\&A_2$			
1		A_0	M_1	M_2	$A_1\&A_2$		
0			A_0	M_1	M_2	$A_1\&A_2$	
1				A_0	M_1	M_2	$A_1\&A_2$

(b)

Figure 1: The graph representation of a toy program (a), where A_i and M_i stand for action and match nodes/operations. Supposing a processor can initiate ≤ 1 match per clock cycle, (b) encodes an optimal embedding of the program, where $P = 2$.

Results for this model. *DPEP under the WIDTH model is $\mathscr{NP}$-hard. Bad news: the optimal number of cores to achieve line rate cannot be approximated better than 3/2 unless $\mathscr{P}$=$\mathscr{NP}$. Good news: the optimum can be 3/2-approximated in polynomial time (see Theorems 4 and 5).*

2.4 WIDTH-IPC1: Full-blown dRMT model

Our next model, WIDTH-IPC1, is equivalent to the one studied in [2]. Here, we simultaneously require IPC= 1 and allow arbitrary table widths. As expected, combining additional constraints does not make the problem easier: the minimal P for which an embedding exists cannot be approximated better than 3/2 (unless $\mathscr{P}$=$\mathscr{NP}$, see Theorem 4). As a promising positive result, though, we show that in WIDTH-IPC1 the optimum can be 4-approximated in polynomial time; see Alg. 1 (see Theorem 9). The algorithm is based on the observation that the optimal period for a scheduling solution (see Definition 1) is independent of values ΔM and ΔA, because it depends only on the *number* of clock cycles with at least one match/action node (see Lemma 8). Our algorithm greedily finds a solution with $\Delta M = \Delta A = 1$ (a *pre-scheduling*, see Definition 6) such that clock cycles are filled with match/action nodes at least half full when possible, resulting a 4-approximation.

Results for this model. *DPEP under the WIDTH-IPC1 model is $\mathscr{NP}$-hard. Bad news: the optimal number of cores to achieve line rate cannot be approximated better than 3/2 unless $\mathscr{P}$=$\mathscr{NP}$. Good news: the optimum can be 4-approximated in polynomial time (see Theorems 4 and 9)*

2.5 WIDTH-IPC2: Loose IPC constraints

The original paper [2] also considers the case when IPC is 2, possibly allowing more compact program embeddings. Intuitively speaking, increasing IPC from 1 to 2 may allow at most twice as efficient embeddings. Thus, the the greedy algorithm of model WIDTH-IPC1 will give an 8-approximation in the WIDTH-IPC2 model.

Results for this model. *DPEP under the WIDTH-IPC2 model is $\mathscr{NP}$-hard. Bad news: the optimal number of cores to achieve line rate cannot be approximated better than 3/2 unless $\mathscr{P}$=$\mathscr{NP}$. Good news: the optimum can be 8-approximated in polynomial time (see Theorems 4 and 13).*

For IPC= 2 the ILP solvers of [2] can compute efficient program embeddings relatively easily. Thus, we will not study this model further here.

3 Main result

3.1 Formal problem statement

Suppose that one of the DPEP model inputs is given with directed acyclic graph $D = (V, E)$ and input parameters $\Delta M, \Delta A, \overline{M}, \overline{A}, IPC \in \{1, 2, \infty\}$ and $W : V \to \mathbb{N}$. For brevity we will use the latency function $l : V \to \{\Delta M, \Delta A\}$, where $l(v) = \Delta M$ or ΔA if v is a match/action node, respectively.

Definition 1. *A **scheduling** of the nodes is a function $S : V \to \mathbb{N}^+$ such that for every arc $(v_i, v_j) \in E$ we have $S(v_j) - S(v_i) \geq l(v_i)$.*

*For a scheduling S and period $P \in \mathbb{N}^+$, let $\mathcal{S}_P$ denote the set of schedulings S_i such that $S_i(v) = S(v) + iP$ (for $i \in \mathbb{N}$). We say that a scheduling S is **feasible with period** P if*

1. $\forall t \in \mathbb{N}^+ : \sum_{S_i \in \mathcal{S}_P} \sum_{\substack{v_m \in V_m \\ S_i(v_m)=t}} W(v_m) \leq \overline{M}$
2. $\forall t \in \mathbb{N}^+ : \sum_{S_i \in \mathcal{S}_P} \sum_{\substack{v_a \in V_a \\ S_i(v_a)=t}} W(v_a) \leq \overline{A}$
3. $\forall t \in \mathbb{N}^+ : \#\{S_i \in \mathcal{S}_P | \exists v_m \in V_m : S_i(v_m) = t\} \leq IPC$
4. $\forall t \in \mathbb{N}^+ : \#\{S_i \in \mathcal{S}_P | \exists v_a \in V_a : S_i(v_a) = t\} \leq IPC.$

*In a **DPEP** instance, the goal is to find the minimum P such that there exists a scheduling S which is feasible with period P. The decision version of DPEP is to decide for a given value k if there exists a feasible P-periodic scheduling with $P \leq k$.*

Model name:	**BASIC**	**IPC1**	**WIDTH**	**WIDTH-IPC1**	**WIDTH-IPC2**
New feature on top of the basic constraints	(basic model)	Max. 1 packet per processor per cycle (IPC= 1)	arbitrary table widths	arbitrary table widths + IPC= 1	arbitrary table widths + IPC= 2 (≤2 pkt./proc./cycle)
Complexity	$\mathscr{P}$	$\mathscr{NP}$-hard	$\mathscr{NP}$-hard	$\mathscr{NP}$-hard	$\mathscr{NP}$-hard
Bad news: Inapproximable better than ... *(,unless $\mathscr{P}=\mathscr{NP}$)*	OPT	4/3·OPT	3/2·OPT	3/2·OPT	3/2·OPT
Good news: Constant approximable in...	OPT	3·OPT	3/2·OPT	4·OPT	8·OPT

Table 1: Overview of the main results. Bad news: the Disaggregated Pipeline Embedding Problem (PEP) is $\mathscr{NP}$-hard even with relaxing some constraints. Good news: the DPEP is polynomially solvable under the BASIC model, and is constant approximable in polynomial time even when considering the model tackled by [2].

3.2 Complexity

Theorem 2. *For model BASIC $P = \max\left(\left\lceil \frac{|V_m|}{\overline{M}} \right\rceil, \left\lceil \frac{|V_a|}{\overline{A}} \right\rceil\right)$ is the optimal period, and a feasible P-periodic scheduling can be found in polynomial time, in $O\big(|E| + |V| + P \log P\big)$.*

PROOF: It is clear that $\left\lceil \frac{|V_m|}{\overline{M}} \right\rceil$ and $\left\lceil \frac{|V_a|}{\overline{A}} \right\rceil$ are lower bounds for P. To prove the other direction, let $v_1, v_2, \ldots, v_n$ be an arbitrary topological order of the nodes (i.e. $i < j$ if $v_i v_j \in E$). We will construct a scheduling S of the nodes in this order in the following way. $S(v_1) := 1$. For $j > 1$, let $\delta_j := \max\{S(v_i) + l(v_i) \mid v_i v_j \in E\}$ if v_j has at least one entering arc, otherwise $\delta_j := S(v_{j-1})$. If $v_j \in V_m$, let $S(v_j) := \min\{k \geq \delta_j \mid \#\{i : i < j, v_i \in V_m, S(v_i) \equiv k \mod P\} \leq \overline{M}\}$. Similarly, if $v_j \in V_a$, let $S(v_j) := \min\{k \geq \delta_j \mid \#\{i : i < j, v_i \in V_a, S(v_i) \equiv k \mod P\} \leq \overline{A}\}$. Note that by the choice of P, the set to be minimized is never empty. The total number of steps for calculating all δ_i values is $O(|E|)$. In order to determine a minimum value for an $S(v_j)$ let us store for every residue class k the next class $n(k)$ which is not full. When a residue class k becomes full we need to union classes pointing to k and $n(k)$. There are at most P union steps and they can be done in a total running time of $O\big(P \log P\big)$. Thus $S(v_i)$ can be determined in $O(1)$ time, giving a total running time of $O\big(|E| + |V| + P \log P\big) = O\big(|E| + |V| \log |V|\big)$. □

Theorem 3. *The decision versions of models IPC1 is NPC. Furthermore, the optimal period cannot be approximated better than a ratio of* $4/3$ *unless* $\mathscr{P}=\mathscr{NP}$. □

The proof reduces $P|prec, p_j = 1|C_{max}$ scheduling problems to IPC1 problem instances.

Theorem 4. *The decision versions of models WIDTH, WIDTH-IPC1 and WIDTH-IPC2 are NPC, and their optimal period cannot be approximated better than a ratio of* $3/2$ *unless* $\mathscr{P}=\mathscr{NP}$. □

The proof relies on reducing `2-PARTITION` problems to WIDTH, WIDTH-IPC1 and WIDTH-IPC2 instances.

3.3 Approximation algorithms

First we give a $3/2$-approximation for model WIDTH by reducing it to bin packing.

Theorem 5. *For model WIDTH, a scheduling having a period at most* $3/2$ *times the optimal can be calculated in* $O(|V| + |E|)$ *time.*

PROOF: First, we ignore the precedence constraints and consider two bin-backing problems with node widths as object weights and $\overline{A}, \overline{M}$ as bin capacities. Using the linear-time $3/2$-approximation algorithm of [1], we separately sort the match and action nodes in a number of M' and A' bins, respectively. Thus, a scheduling with a period $P = \max\{M', A'\}$ would be a $3/2$-approximation on the optimal period. Now we show that such a scheduling exists. Let B_m^i and B_a^i denote the i^{th} bin of match and action nodes, respectively. For each $i \in \{1, \ldots, P\}$, we assign B_m^i and B_a^i to residue class i. Now, we take into count the precedence constraints again. Then, we take an arbitrary topological order of the nodes $v_1, \ldots, v_n$. For each $j \in \{1, \ldots, n\}$, we schedule v_j (being part of a batch B_v^i) to the smallest positive integer clock cycle $S(v_j)$ that is $\equiv i \mod P$, and is greater or equal with $S(v) + l(v)$, for each node v directly preceding v_j in the ODG (that is, $(v, v_j) \in E$). Note that if v_j does not have any in-arc, it is scheduled to clock cycle i. The proof follows. □

Now we describe approximation algorithms for models IPC1, WIDTH-IPC1, and WIDTH-IPC2. The key idea is to find a proper partial order of the nodes that can be expanded into a scheduling.

Definition 6. *A function* $PS : V \to \mathbb{N}^+$ *is a* ***pre-scheduling****, if*

1. $PS(v_m) \neq PS(v_a)$ *for every* $v_m \in V_m, v_a \in V_a$,
2. $PS(v_j) - PS(v_i) \geq 1$ *for every arc* $(v_i, v_j) \in E$,
3. *if* $PS^{-1}(k) = \emptyset$ *for a* $k \in \mathbb{N}^+$, *then* $PS(v) < k$ *for every* $v \in V$.
4. $\forall t \in \mathbb{N}^+ : \sum_{\substack{v_m \in V_m \\ PS(v_m)=t}} W(v_m) \leq \overline{M}$,

5. $\forall t \in \mathbb{N}^+ : \sum_{\substack{v_a \in V_a \\ PS(v_a)=t}} W(v_a) \leq \overline{A}$.

Let $L(PS)$ denote the ***length*** *of the pre-scheduling, so the largest clock cycle that has an embedded node:*

$$L(PS) = \max\{i | PS^{-1}(i) \neq \emptyset\}.$$

Let A denote the number of clock cycles with at least one embedded action node. Formally, $A := \#\{i \in \mathbb{N}^+ | PS^{-1}(i) \cap V_a \neq \emptyset\}$. We define M similarly with match nodes.

A scheduling S is an ***expansion of a pre-scheduling*** *PS if there exists a strictly monotone function $f : \mathbb{N}^+ \to \mathbb{N}^+$ such that $S(v) = f(PS(v))$.*

Claim 7. *Every pre-scheduling has an expansion.*

PROOF: We determine values $f(1), \ldots, f(L(PS))$ in this order. Let $f(1) = 1$. For $1 < i \leq L(PS)$, if there is no arc entering nodes in $PS^{-1}(i)$, then $f(i) := f(i-1)+1$. Else let $f(i) := \max\{f(PS(v)) + l(v) | vw \in E, PS(w) = i\}$. □

Lemma 8. *Let PS be a pre-scheduling and let $IPC = 1$. For $P := \max(A, M)$ there exists an expansion of PS that is feasible with period P. Moreover, P is the smallest among such periods.*

PROOF: It is easy to see that values A and M are lower bounds for the period of an expansion because the resulting scheduling has the same number of match/action clock cycles.

Now we show that PS has a feasible P-periodic expansion.

We have seen in Claim 7 that PS has an expansion. We use a similar approach to get a feasible P-periodic scheduling. In addition, we will make sure that there are no two clock cycles with the same type of nodes embedded into the same residue class modulo P, which will guarantee constraints (1)-(4) of a feasible scheduling.

Let $f(1) = 1$ and for $1 < i \leq L(PS)$ we do the followings. If there exists an arc entering a node in $PS^{-1}(i)$ then $\delta := \max\{f(PS(v)) + l(v) | vw \in E, PS(w) = i\}$, otherwise $\delta := f(i-1)+1$.

$$f(i) := \min\{k \geq \delta \mid \nexists j < i : f(j) \equiv k \mod P \text{ and } PS^{-1}(i), PS^{-1}(j) \text{ have the same type }\} \quad (1)$$

Note that the set we are minimizing for $f(i)$ is not empty since $P \geq M$ and $P \geq A$, and former clock cycles of the same type cannot cover all residue classes modulo P. □

Theorem 9. *There is a 4-approximation algorithm for model WIDTH-IPC1.*

PROOF: Based on Lemma 8, our goal is to find a pre-scheduling PS where we minimize $\max(A, M)$. Our algorithm uses a greedy approach and embeds at least half full clock cycles as long as it is possible (see Algorithm 1).

The algorithm maintains the subset V' of nodes that need to be embedded. At the beginning, let $V' := V$. Let m/a denote the current list of match/action nodes of zero indegree in the subgraph spanned by V', sorted in a descending order according to their width. At one phase of the algorithm, we embed some nodes from m and a to one or two clock cycles and then move to the next clock cycle and the next phase, when m and a are updated again. Let i denote the current first empty clock cycle.

Let w_m and w_a denote the sum of widths of nodes in m and a, respectively. In one phase, we do the following:

If $w_m < \overline{M}/2$ and $w_a < \overline{A}/2$, we embed all nodes in m to clock cycle i and all nodes in a to clock cycle $i+1$. We move on to the next clock cycle: $i := i+2$.

If $w_m \geq \overline{M}/2$ and $w_a < \overline{A}/2$, we greedily embed only nodes in m in clock cycle i as long as possible, and move to the next clock cycle: $i = i+1$.

Similarly, if $w_m < \overline{M}/2$ and $w_a \geq \overline{A}/2$, we greedily embed nodes in a in clock cycle i as long as possible, and then move on to the next phase and the next clock cycle.

Algorithm 1: WIDTH-IPC1 Our Greedy

Input: ODG $D = (V, E)$; $W : V \to \mathbb{N}^+$; $\overline{M}, \overline{A}$
Output: $PS : V \to \mathbb{N}^+$
begin
1 $i := 1$; $V' := V$
2 **while** $V' \neq \emptyset$ **do**
3 $a :=$ list of action nodes with 0 indegrees, descending order of width
4 $m :=$list of match nodes with 0 indegrees, descending order of width
5 $w_a :=$ sum of widths in a
6 $w_m :=$ sum of widths in m
7 current_usage := 0
8 **if** $w_m \geq 1/2\overline{M}$ *and* $w_a \geq 1/2\overline{A}$ **then**
9 Go to line 12 or 19
10 **if** $w_a \geq 1/2\overline{A}$ *and* $w_m < 1/2\overline{M}$ **then**
11 Go to line 19
12 **while** $m[0]$+*current_usage* $\leq \overline{M}$ **do**
13 current_usage += $m[0]$
14 PS$[m[0]] := i$
15 $V' := V' \setminus \{m[0]\}$
16 $m := m - m[0]$
17 i := i+1
18 **if** $w_m \geq 1/2\overline{M}$ **then**
continue
19 **while** *a[0]*+*current_usage* $\leq \overline{A}$ **do**
20 current_usage += $a[0]$
21 PS$[a[0]] := i$
22 $V' := V' \setminus \{a[0]\}$
23 $a := a - a[0]$
24 $i := i + 1$
25 **return** PS

Finally, if both $w_m \geq \overline{M}/2$ and $w_a \geq \overline{A}/2$, we can choose m or a arbitrarily and embed nodes again greedily as before into clock cycle i.

Now we prove 4-approximation. We partition the clock cycles into four groups: let HF_m/HF_a denote those clock cycles that are at least half full with match/action nodes, respectively, and similarly, let NHF_m/NHF_a denote the list of those that are not half full. Note that $|NHF_m| = |NHF_a|$ and $NHF_m[j] = NHF_a[j] - 1$. We can assume that $M \geq A$. From Lemma 8, we get that the constructed pre-scheduling can be expanded into feasible scheduling with period M. Let P_o denote the optimal period for the problem. We know that $P_o \geq \sum_{v \in V_m} W(v)/\overline{M}$. Since $\sum_{v \in V_m} W(v)/\overline{M} \geq \sum_{v \in PS^{-1}(HF_m)} W(v)/\overline{M} \geq |HF_m|/2$ and we get $|HF_m| \leq 2P_o$.

Claim 10. *For every node v embedded in $NHF_m[i]$ or $NHF_a[i]$ ($i \geq 2$) there is a path from a node embedded in $NHF_m[i-1]$ or $NHF_a[i-1]$ to v.*

Proof: Let us consider the phase when $m = NHF_m[i-1]$ and $a = NHF_a[i-1]$. Observe that every node in the current V' is reachable from nodes embedded in $NHF_m[i-1]$ or $NHF_a[i-1]$. □

Claim 11. *There is a path of length of at least $|NHF_m|$ in D.*

Proof: Applying Claim 10 backwards starting from an arbitrary node $v_{|NHF_m|}$ embedded in $NHF_m[|NHF_m|]$ or $NHF_a[|NHF_m|]$ we get a path from a node $v_{|NHF_m|-1}$ embedded in $NHF_m[|NHF_m|-1]$ or $NHF_a[|NHF_m|-1]$, and so on. By concatenating these paths, we get a path $\mathcal{P}$ which is required in the claim. □ Note that for any path Q we have that $|V(Q) \cap V_m| \leq P_o$ and $|V(Q) \cap V_a| \leq P_o$ so $|V(Q)| \leq 2P_o$. Hence $|NHF_m| \leq |V(\mathcal{P})| \leq 2P_o$ and so $|M| = |HF_m| + |NHF_m| \leq 4P_o$, which proves the theorem. The running time of the algorithm is $O(|V| \log |V| + |E|)$.

□

Theorem 12. *Model IPC1 can be* 3−*approximated in polynomial time.*

PROOF:[Sketch of proof] We can simplify the previous algorithm in Theorem 9 the following way: we do not need to sort the elements in lists m and a because all have unit width. Moreover, we apply limits $\overline{M}$ and $\overline{A}$ for embeddings instead of $\overline{M}/2$ and $\overline{A}/2$, and embed nodes to get full clock cycles (with either match or action nodes only). Let F_m and F_a denote the set of full clock cycles. We can derive a sharper bound $|F_m| \leq P_o$, which gives a 3-approximation. □

Finally, we can derive an approximation algorithm for the WIDTH-IPC2 model from the one given for the WIDTH-IPC1 .

Theorem 13. *Model WIDTH-IPC2 can be* 8−*approximated in polynomial time.*

PROOF: Let P_1^{opt} and P_2^{opt} denote the optimal periods for WIDTH-IPC1 and WIDTH-IPC2, respectively for a pair of models with the same input parameters (except IPC). Since a feasible P-periodic one for WIDTH-IPC2 can be transformed into a feasible $2P$-periodic scheduling for WIDTH-IPC1, so $P_1^{opt} \leq 2P_2^{opt}$. Let P^* denote the period of the scheduling given by the 4−approximation algorithm for WIDTH-IPC1 (Theorem 9). Then $P^* \leq 4P_1^{opt} \leq 8P_2^{opt}$. □

4 Simulation results

In this section, we present our simulation studies on P4 embeddings for the dRMT architecture over the WIDTH-IPC1 model. Our goal is to maximize throughput while keeping latency under control. Running times were measured on a commodity laptop, with 64 GB RAM and 24 threads, at 2.40GHz. The code used in the evaluation is available on GitHub (`https://github.com/fraknoiadam/drmt`).

Maximizing the throughput. The throughput of a dRMT switch is inversely proportional to the number of processors P needed to achieve line rate [2]. Table 2 summarizes the lowest P values computed by different algorithms. In summary, Alg. 1 uses at most 19% more processors than the best ILP solution, compared to the at most $> 36\%$ extra processors used by the heuristic rnd_sieve of [2]. Recall, Alg. 1 is a *provably* constant approximation on the optimal P. In addition, the running time of Alg. 1 on the "Egress", "Ingress", and "Combined" instances obtained from `switch.p4` [2] was 7 ms, 24 ms, and 41 ms, respectively, which is beyond an order of magnitude improvement over rnd_sieve [2]. The average running time (to achieve their best results) of Alg. 1 and rnd_sieve on these graphs were 0.007, 0.28, 10.5 and 0.3, 1.5, 2.7 `[sec]`, respectively. Out of 1000 runs, Alg. 1 reached the theoretically optimal P values 1000, 85, and 4 times, exploiting the fact that, in Alg. 1 there are multiple steps where random choices are made (e.g., at lines 3 and 4).

Algorithm \ Graph	Egress $\lvert V\rvert = 104$ $\lvert E\rvert = 291$	Ingress $\lvert V\rvert = 224$ $\lvert E\rvert = 930$	Combined $\lvert V\rvert = 328$ $\lvert E\rvert = 1221$
rnd_sieve i.e., [2]-greedy	13	21	30
Our greedy	13	19	23
[2] ILP	11	17	21
ILP lower bound	7	15	21

Table 2: Best P values computed by different algorithms

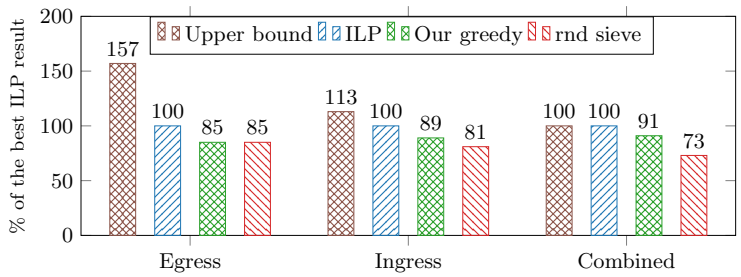


Figure 2: Throughput provided by different heuristics as precentage of the best ILP solution

Fig. 2 visualizes the throughput provided by our greedy algorithm, and rnd_sieve [2] as the percentage of the best throughput provided by the optimal ILPs. For the Egress, Ingress, and Combined instances, Alg. 1 achieves 85%, 89%, and 91%, while rnd_sieve yields 85%, 81%, and 73%, respectively. In other words, our algorithm performs at least as well as the rnd_sieve. Moreover, in these cases, with the size of the input graph growing, Alg. 1 got closer to the best throughput computed by the ILP formulation, while the relative performance of rnd_sieve degraded.

	Egress			Ingress			Combined		
P	11	12	13	17	18	19	21	22	23
optimal T_P	217	208	206	245	246	244	243	244	243
time [sec]	1203	76	107	106	59	23	118	25	109

Table 3: P vs T: higher throughput does not mean higher latency.

Running times and latency. For each of the three program instances, Table 3 shows the optimal latency ($T = T_P$) in the case of the three lowest P values that could be computed by the ILPs of the paper. We can see that, while approaching the best P obtained by the ILP, the optimal latency remained more or less steady in the case of Ingress and Combined, and, for Egress, it grew only by less than 5% also. Running times for achieving the optimal T values did not grow radically either on the example cases, except for Egress. We can conclude that, contrary to the intuition, a higher throughput (i.e., a lower P) has no significant impact on the lowest latency (T) achievable.

References

[1] Rudolf Berghammer and Florian Reuter. A linear approximation algorithm for bin packing with absolute approximation factor 32. *Science of Computer Programming*, 48(1):67–80, 2003.

[2] Sharad Chole, Andy Fingerhut, Sha Ma, Anirudh Sivaraman, Shay Vargaftik, Alon Berger, Gal Mendelson, Mohammad Alizadeh, Shang-Tse Chuang, Isaac Keslassy, et al. dRMT: Disaggregated Programmable Switching. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, pages 1–14, 2017.

[3] Balázs Vass, Ádám Fraknói, Erika Bérczi-Kovács, and Gábor Rétvári. Compiling packet programs to drmt switches: Theory and algorithms. In *Proceedings of the 5th International Workshop on P4 in Europe*, EuroP4 '22, page 26–32, New York, NY, USA, 2022. Association for Computing Machinery.

Algebraic realizations of pairs of closure operators

Dániel Garamvölgyi

Department of Operations Research
Eötvös Loránd University
daniel.garamvolgyi@ttk.elte.hu

Abstract: Given a field extension $K \subseteq L$ and a finite set of elements $E \subseteq L$, there are at least two natural ways to define a closure operator on E: for a subset $X \subseteq E$ we can either take the subset of elements in E that are algebraically dependent on X over K, or the subset of elements that are generated by X over K. It is well-known that the former of these defines a matroid; matroids defined in this way are said to be *algebraic over* K. On the other hand, the combinatorial structure of the latter type of closure operator seems to be largely unexplored, despite the fact that in some applications (e.g., rigidity theory and low-rank matrix completion) both closure operators encode relevant information.

In this note we initiate the systematic investigation of the following question: which pairs of closure operators $(\mathrm{cl}, \mathrm{scl})$ correspond to a pair of closure operators defined in the above manner? We give natural necessary conditions and consider in detail the cases $\mathrm{scl} = \mathrm{id}_E$ and $\mathrm{scl} = \mathrm{cl}$.

Keywords: algebraic matroid, closure operator, rigidity theory

1 Closure operators and algebraic matroids

Throughout this note we assume basic familiarity with matroid theory and field theory. For detailed introductions, see [7] and [5], respectively.

We shall be concerned with closure operators arising in the study of algebraic matroids. Let E be a set and $\mathrm{cl} : \mathcal{P}(E) \to \mathcal{P}(E)$ a function on the subsets of E. We say that cl is a *closure operator* if it satisfies the following for every pair of subsets $X, Y \subseteq E$:

i) *(Monotone)* $X \subseteq \mathrm{cl}(X)$,

ii) *(Inflationary)* $X \subseteq Y \Rightarrow \mathrm{cl}(X) \subseteq \mathrm{cl}(Y)$,

iii) *(Idempotent)* $\mathrm{cl}(\mathrm{cl}(X)) = \mathrm{cl}(X)$.

We say that two closure operators cl and cl' defined on sets E and E', respectively, are *isomorphic* if there is a bijection $f : E \to E'$ of the ground sets such that for every subset $X \subseteq E$ we have $\mathrm{cl}(f(X)) = f(\mathrm{cl}'(X))$. In other words, isomorphic closure operators are "the same up to a relabeling of their ground sets." In this case we say that f is an *isomorphism* between cl and cl'.

Let cl be a closure operator on a finite set E. We say that cl is a *matroid closure operator* if it satisfies the following for every pair of elements $x, y \in E$ and subset $X \subseteq E$:

iv) *(Mac Lane-Steinitz exchange property)* If $x, y \notin \mathrm{cl}(X)$ and $y \in \mathrm{cl}(X + x)$, then $x \in \mathrm{cl}(X + y)$.

A matroid closure operator induces a matroid on E in which the independent sets are those subsets $X \subseteq E$ that satisfy $x \notin \mathrm{cl}(X - x)$ for every $x \in X$. Two matroid closure operators are isomorphic precisely if the matroids induced in this way are isomorphic.

Algebraic matroids encode the combinatorial structure of algebraic dependence over some field. Let K be a field and $K \subseteq L$ a field extension. For a subset $X \subseteq L$, let $K(X)$ denote the subfield of L generated by the elements in $K \cup X$. Given a finite subset $E \subseteq L$, we define a closure operator cl_K^E by letting

$$\mathrm{cl}_K^E(X) = \{x \in E : x \text{ is algebraic over } K(X)\}$$

for each $X \subseteq E$. It is well-known that cl_K^E is a matroid closure operator, see, e.g., [7, Section 6.7]. We say that the matroid induced by cl_K^E is the *algebraic matroid associated to E over K*. We say that a matroid $\mathcal{M}$ is *algebraic over a field K* if there is some field extension $K \subseteq L$ and a finite subset $E \subseteq L$ such that the matroid associated to E over K is isomorphic to $\mathcal{M}$.

Example 1 *Let $K_V = (V, E)$ be the complete graph on vertex set V and let $d \geq 1$ be an integer. The d-dimensional generic rigidity matroid of K_V is the algebraic matroid over $\mathbb{R}$ associated to the polynomials*

$$f_{uv} = \sum_{i=1}^{d} (x_u^i - x_v^i)^2, \qquad \forall uv \in E,$$

where $x_v^i, i \in \{1 \ldots, d\}, v \in V$ are independent transcendentals over $\mathbb{R}$. It is well-known that changing the base field from $\mathbb{R}$ to $\mathbb{C}$ in this definition gives the same matroid.

If the base field K is algebraically closed, then algebraic matroids can also be viewed in geometric terms, which we outline next. Let us consider the vector space K^E whose axes are labeled by elements of E. A set $Z \subseteq K^E$ is a *variety* if it is the set of simultaneous zeros of some polynomials in $|E|$ variables. A variety is *irreducible* if it cannot be written as the union of two varieties properly contained in it. Given an irreducible variety $Z \subseteq K^E$ and a subset $X \subseteq E$ of the ground set, let $Z_X \subseteq K^X$ denote the projection of Z onto the axes corresponding to X. For each pair of subsets $X \subseteq X' \subseteq E$, we can define the projection $\pi_{X',X} : Z_{X'} \to Z_X$. With this notation in place, we may define a matroid closure operator on E associated to Z by setting

$$\mathrm{cl}(X) = \{x \in E : \pi_{X+x,X}^{-1}(z) \text{ is finite for almost all } z \in Z_X\}$$

That is, for almost all[1] points $z \in Z_X$, there are only finitely many points $z' \in Z_{X+x}$ that project to z. By basic results of algebraic geometry, this defines a matroid closure operator, and the matroids obtained in this way are precisely the algebraic matroids over K (keeping in mind the assumption that K is algebraically closed). For a detailed exposition of this correspondence, see [8].

Example 2 *Let us return to the generic d-dimensional rigidity matroid of the complete graph $K_V = (V, E)$. For a pair of points $p, q \in \mathbb{C}^d$ with $p = (p^1, \ldots, p^d)$ and $q = (q^1, \ldots, q^d)$, let us define their complex squared distance to be $\sum_{i=1}^{d} (p^i - q^i)^2 \in \mathbb{C}$. Let $Z \subseteq \mathbb{C}^E$ be the set of vectors that can be obtained as the vector of pairwise squared distances of some configuration $p_v \in \mathbb{C}^d, v \in V$. It is known that Z is an irreducible variety, and the associated matroid is precisely the d-dimensional generic rigidity matroid on K_V. Geometrically, an edge $u'v' \in E$ is in the closure of a set of edges $X \subseteq E$ if for almost all vectors $(\ell_{uv})_{uv \in X} \in Z_X$ there are only a finite number of possible values $\ell_{u'v'} \in \mathbb{C}$ such that there is a configuration $p_v \in \mathbb{C}^d, v \in V$ in which the squared distance of p_u and p_v is ℓ_{uv} for each $uv \in X \cup \{u'v'\}$.*

Example 3 *(Following [6]) Let $Z \subseteq \mathbb{C}^{n \times n}$ be the set of matrices of rank at most r. It is well-known that Z is an irreducible variety. The matroid associated to Z on $E = \{1, \ldots, n\}^2$ is called the rank-r determinantal matroid. An element $(i', j') \in E$ is in the closure of a set $X \subseteq E$ in this matroid if for almost all collections $(\alpha_{ij})_{(i,j) \in X} \in Z_X$, there are only a finite number of possible values $\alpha_{i'j'} \in \mathbb{C}$ for which there is a matrix $M \in \mathbb{C}^{n \times n}$ of rank at most r such that $M_{ij} = \alpha_{ij}$ for each $(i, j) \in X \cup \{(i', j')\}$.*

As we shall see in the next section, there is another closure operator associated to finitely generated field extensions (or irreducible varieties) that can be defined naturally both from the algebraic and the geometric viewpoints.

[1] By "almost all", we mean that the points in Z_X not having this property are contained in a variety that does not contain all of Z_X.

2 The strong closure operator and its basic properties

Given a field extension $K \subseteq L$ and a finite subset $E \subseteq L$, we can define another closure operator by letting

$$\mathrm{scl}_K^E(X) = \{x \in E : x \in K(X)\}$$

for each $X \subseteq E$. We call this the *strong closure* operator associated to E over K. It is easy to verify that scl_K^E is indeed a closure operator. However, in contrast with cl_K^E, scl_K^E may not be a matroid closure operator, as the following simple example shows.

Example 4 *Let x be transcendental over $\mathbb{C}$ and let $E = \{x, x^2\}$. We have $\mathrm{scl}_{\mathbb{C}}^E(\{x\}) = \{x, x^2\}$ while $\mathrm{scl}_{\mathbb{C}}^E(\{x^2\}) = \{x^2\}$. This shows that $\mathrm{scl}_{\mathbb{C}}^E$ does not satisfy the Mac Lane-Steinitz exchange property for x, x^2 and $\varnothing$.*

Over an algebraically closed field K, the strong closure can also be defined in the geometric setting outlined in the previous section. Using the same notation, the strong closure on a set E associated to an irreducible variety $Z \subseteq K^E$ can be defined by

$$\mathrm{scl}(X) = \{x \in E : |\pi^{-1}_{X+x,X}(z)| = 1 \text{ for almost all } z \in Z_X\}$$

That is, for almost all points $z \in Z_X$, there is a unique point $z' \in Z_{X+x}$ that projects to z.

Example 5 *In the case of the d-dimensional generic rigidity matroid on $K_V = (V, E)$, an edge $u'v'$ is in the strong closure of $X \subseteq E$ if for almost all values $(\ell_{uv})_{uv \in X} \in Z_X$, the squared distance of $p_{u'}$ and $p_{v'}$ is the same in each configuration $p_v \in \mathbb{C}^d, v \in V$ in which the squared distance of p_u and p_v is ℓ_{uv} for each $uv \in X$. Algebraically, this means that the "edge length polynomial" $f_{u'v'}$ (as defined in Example 1) is contained in the field $\mathbb{C}(\{f_{uv}, uv \in X\})$. In the rigidity theory literature this is referred to as "$\{u', v'\}$ being globally linked in $\mathbb{C}^d$ in the graph $G = (V, X)$", see [3].*

Example 6 *In the case of the rank-r determinantal matroid, (i', j') is in the strong closure of $X \subseteq \{1, \ldots, n\}^2$ if for almost all collections $(\alpha_{ij})_{(i,j)\in X} \in Z_X$, the value at the $i'j'$-th position is the same in each matrix $M \in \mathbb{C}^{n\times n}$ of rank at most r that satisfies $M_{ij} = \alpha_{ij}$ for all $(i, j) \in X$. The same notion appears in [6] under the name "unique closure".*

Our aim in this note is to gain some understanding of the combinatorial properties of scl_K^E, and in particular of the relationship between cl_K^E and scl_K^E. To make this more precise, we introduce the following notion. Let E' be a finite set and let $(\mathrm{cl}, \mathrm{scl})$ be a pair of closure operators on E'. Let K be a field. We say that the pair $(\mathrm{cl}, \mathrm{scl})$ is *algebraically realizable over K* if there is some field extension $K \subseteq L$, a finite subset $E \subseteq L$ and a bijection $f : E' \to E$ such that f is both an isomorphism between cl_K^E and cl, and an isomorphism between scl_K^E and scl. In this case, we say that E *algebraically realizes* the pair $(\mathrm{cl}, \mathrm{scl})$ over K.

Clearly, for $(\mathrm{cl}, \mathrm{scl})$ to be algebraically realizable over any field, cl needs to be a matroid closure operator. It is also immediate that we must have

$$\mathrm{scl}(X) \subseteq \mathrm{cl}(X) \qquad \forall X \subseteq E. \tag{*}$$

The following lemma gives another necessary condition for the existence of an algebraic realization.

Lemma 7 *Let* cl *and* scl *be closure operators on a finite set E such that* $(\mathrm{cl}, \mathrm{scl})$ *is algebraically realizable over a field K. Then the following holds for every $x, y \in E$ and $X \subseteq E$:*

("mixed exchange property") if $x \notin \mathrm{scl}(X), y \notin \mathrm{cl}(X)$ and $x \in \mathrm{scl}(X + y)$, then $y \in \mathrm{cl}(X + x)$. (**)

Proof: By passing to an algebraic realization we may assume that $E \subseteq L$ for some field extension $K \subseteq L$ and that $\mathrm{cl} = \mathrm{cl}_K^E$ and $\mathrm{scl} = \mathrm{scl}_K^E$. Note that $y \notin \mathrm{cl}(X)$ says that y is transcendental over $K(X)$,

so $K(X+y)$ is isomorphic to the field of fractions of the polynomial ring $K(X)[t]$. Since $x \in \mathrm{scl}(X+y)$, there are polynomials $f, g \in K(X)[t]$ such that $x = \frac{f(y)}{g(y)}$. Furthermore, since $x \notin \mathrm{scl}(X)$, at least one of f and g is nonconstant. It follows that the polynomial $f(t) - xg(t)$ is nonconstant and has y as one of its zeros. This shows that y is algebraic over $K(X+x)$, which is equivalent to $y \in \mathrm{cl}(X+x)$. $\square$

Note that if $\mathrm{scl} = \mathrm{id}_E$, then (**) is satisfied trivially, while if $\mathrm{scl} = \mathrm{cl}$, then (**) is precisely the Mac Lane-Steinitz exchange property for cl. It is unclear whether there are any other necessary conditions for the existence of an algebraic realization besides (*) and (**). We go out on a limb and make the following conjecture.

Conjecture 8 *Let $\mathcal{M} = (E', \mathcal{I})$ be a loopless matroid that is algebraic over the field K. Let $\mathrm{cl}_{\mathcal{M}}$ be the closure operator of $\mathcal{M}$ and let* scl *be a closure operator on E'. If the pair $(\mathrm{cl}_{\mathcal{M}}, \mathrm{scl})$ satisfies (*) and (**), then $(\mathrm{cl}_{\mathcal{M}}, \mathrm{scl})$ is algebraically realizable over K.*

If we allow loops in the matroid, then Conjecture 8 is false, as shown by the following example.

Example 9 *Consider the matroid $\mathcal{M} = (E', I)$ consisting of a single loop, and let $\mathrm{cl}_{\mathcal{M}}$ denote its closure operator. An algebraic realization E of $\mathcal{M}$ over $\mathbb{C}$ consists of a single element x that is algebraic over $\mathbb{C}$. Since $\mathbb{C}$ is algebraically closed, we must have $x \in \mathbb{C}$. It follows that $\mathrm{scl}_{\mathbb{C}}^{E}(\varnothing) = E$ holds in any algebraic realization. This shows that the pair $(\mathrm{cl}_{\mathcal{M}}, \mathrm{id}_{E'})$ is not realizable over $\mathbb{C}$, even though it satisfies the conditions of Conjecture 8.*

In the rest of this note we examine some special cases of Conjecture 8. More precisely, we shall look at the "edge cases" $\mathrm{scl} = \mathrm{id}_E$ and $\mathrm{scl} = \mathrm{cl}_{\mathcal{M}}$.

3 The edge cases

We first consider the case when $\mathrm{scl} = \mathrm{id}_E$. Our main result is that in this case Conjecture 8 has an affirmative answer (see Theorem 13 below). Our proof is based on the following notion. Let $K \subseteq L$ be fields and let $E \subseteq L$ be a finite subset. Let $\overline{L}$ denote the algebraic closure of L. We say that a set $E' = \{f_x, x \in E\} \subseteq \overline{L} - K$ is a *local modification of E* if f_x is algebraic over $K(x)$ for each $x \in E$.

Lemma 10 *Let $K \subseteq L$ be a field extension and $E \subseteq L$ a finite subset. If $E' = \{f_x, x \in E\}$ is a local modification of E, then $\mathrm{cl}_K^{E'}$ is isomorphic to cl_K^E.*

Proof: This follows from the observation that in the algebraic matroid induced by $E \cup E'$ (as a multiset) over K, x and f_x are parallel elements for each $x \in X$. $\square$

The idea of the proof of Theorem 13 below is that given an algebraic representation E, we can replace each $x \in E$ with one of its p_x-th roots for some sufficiently large prime p_x. This is a local modification, and it can be shown that if we choose the primes $p_x, x \in E$ appropriately, the resulting set E' will be an algebraic realization of $(\mathrm{cl}_K^E, \mathrm{id}_E)$. We shall describe this construction in somewhat more generality. First, we need the following result on radical extensions.

Lemma 11 *[2, Theorem 3.1] Let K be a field, $x \in K$ a nonzero element and p a prime number. The polynomial $t^p - x \in K[t]$ is irreducible over K if and only if it has no roots in K.*

Given a field extension $K \subseteq L$, we use $[L : K]$ to denote the degree of the field extension, i.e., the dimension of L over K as a vector space. Recall that if $K \subseteq L \subseteq L'$ and $[L' : K] < \infty$, then the product formula $[L' : K] = [L : K] \cdot [L' : L]$ holds. In particular, this implies that if $[L : K]$ is finite and $x \in L$, then $[K(x) : K]$ divides $[L : K]$.

Lemma 12 *Let $K \subseteq L$ be fields, $E \subseteq L$ a finite subset and $y \in E$. Suppose that y is transcendental over K. Consider the following closure operator on E:*

$$\mathrm{scl}'(X) = \begin{cases} \mathrm{scl}_K^E(X) - y & y \notin X, \\ \mathrm{scl}_K^E(X) & y \in X. \end{cases}$$

Then the pair $(\mathrm{cl}_E^K, \mathrm{scl}')$ is realizable over K.

Proof: We may assume $L = K(E)$. Let $B \subseteq E$ be a transcendence basis of L over K with $y \in B$. Each element of $E - B$ is algebraic over $K(B)$, so $K(B) \subseteq L$ is a finitely generated algebraic extension, and in particular $n = [L : K(B)]$ is finite. Let $\overline{L}$ denote the algebraic closure of L and let $p > n$ be a prime. Finally, let f_y be an arbitrary p-th root of y in $\overline{L}$.

We claim that $E' = X - y + f_y$ realizes $(\mathrm{cl}_K^E, \mathrm{scl}')$ over K. Clearly, E' is a local modification of E, so cl_K^E and $\mathrm{cl}_K^{E'}$ are isomorphic by Lemma 10. We shall show that scl_K^E is isomorphic to scl'. First observe that the polynomial $t^p - y \in L[t]$ has no roots in L. Indeed, it has no roots in the purely transcendental extension $K(B)$ of K, so by Lemma 11 it is irreducible over $K(B)$. It follows that if $\alpha \in \overline{L}$ is a root, then we have $[K(B)(\alpha) : K(B)] = p > n = [L : K(B)]$. This implies that $\alpha \notin L$, as desired. Using Lemma 11 again we get that $t^p - y$ is irreducible over L, so $[L(f_y) : L] = p$. In particular, $f_y \notin L$.

First consider a subset $X \subseteq E$ with $y \notin X$. Since $K(X) \subseteq L$ we have $f_y \notin K(X)$, so $\mathrm{scl}_K^{E'}(X) = \mathrm{scl}_K^E(X) - y$, as required. Now consider a subset $X \subseteq E$ with $y \in X$, and let $X' = X - y + f_y$. Since $f_y^p = y$, we have $K(X') = K(X)(f_y)$, which implies $\mathrm{scl}_K^E(X) - y + f_y \subseteq \mathrm{scl}_K^{E'}(X')$. Moreover, since $t^p - y$ has no roots in L, it is irreducible over $K(X)$ by Lemma 11, so we have $[K(X') : K(X)] = p$. Suppose for a contradiction that there is some $x \in E$ with $x \in K(X') - K(X)$. Then $[K(X + x) : K(X)]$ divides $[K(X') : K(X)] = p$, but since p is a prime this can only happen if $[K(X + x) : K(X)] = p$ and thus $K(X') = K(X + x)$. In particular this would mean that $f_y \in K(X + x)$, contradicting $f_y \notin L$. This shows that $\mathrm{scl}_K^E(X) - y + f_y = \mathrm{scl}_K^{E'}(X')$, as desired. □

Theorem 13 *Let $K \subseteq L$ be fields and let $E \subseteq L$ be a finite subset. Suppose that every element of E is transcendental over K, so that the matroid induced by cl_K^E is loopless. Then $(\mathrm{cl}_K^E, \mathrm{id}_E)$ is algebraically realizable over K.*

Proof: This follows from Lemma 12 by applying it iteratively to each $y \in E$. □

Next, we consider the algebraic realizability of $(\mathrm{cl}_\mathcal{M}, \mathrm{cl}_\mathcal{M})$, where $\mathrm{cl}_\mathcal{M}$ is a matroid closure operator. We shall use the following special case of the so-called Jacobian criterion for algebraic independence.

Theorem 14 *(Special case of [1, Theorem 8]) Let K be a field and let $f_1, \ldots, f_m \in K[x_1, \ldots, x_k]$ be polynomials of degree 1. The transcendence degree of $\{f_1, \ldots, f_m\}$ over K is equal to the rank of $A = (a_{ij})_{ij} \in K^{m \times k}$, where a_{ij} is the coefficient of x_j in f_i.*

Theorem 15 *Let $\mathcal{M} = (E, I)$ be a matroid and let $\mathrm{cl}_\mathcal{M}$ denote its closure operator. Let K be a field. If $\mathcal{M}$ is linear over K, then the pair $(\mathrm{cl}_\mathcal{M}, \mathrm{cl}_\mathcal{M})$ is algebraically realizable over K.*

Proof: Suppose that $\mathcal{M}$ is on n elements and let $A \in K^{n \times m}$ be a matrix whose columns give a representation of $\mathcal{M}$ over K. Consider the polynomials $f_1, \ldots, f_n \in K[x_1, \ldots, x_m]$ defined by

$$f_i(x_1, \ldots, x_m) = \sum_{j=1}^{m} a_{ij} x_j.$$

It follows from Theorem 14 that the algebraic matroid associated to $E' = \{f_1, \ldots, f_m\}$ over K is the same as the linear matroid defined by the columns of A, which is $\mathcal{M}$. If f_j is algebraically dependent on

$f_i, i \in I$, then there are scalars $\alpha_i, i \in I$ such that $f_j = \sum_{i \in I} \alpha_i f_i$ and thus $f_j \in K(\{f_i, i \in I\})$. This shows that $\mathrm{cl}_K^{E'} = \mathrm{scl}_K^{E'}$, as desired. $\square$

Combining Theorem 15 with the well-known result that over a field of characteristic zero every algebraic matroid is linear, we obtain the following special case of Conjecture 8.

Corollary 16 *Let $\mathcal{M} = (E, I)$ be a matroid and let $\mathrm{cl}_{\mathcal{M}}$ denote its closure operator. Let K be a field of characteristic zero. If $\mathcal{M}$ is algebraic over K, then the pair $(\mathrm{cl}_{\mathcal{M}}, \mathrm{cl}_{\mathcal{M}})$ is algebraically realizable over K.*

It is unclear whether $(\mathrm{cl}_{\mathcal{M}}, \mathrm{cl}_{\mathcal{M}})$ can have an algebraic realization over K if $\mathcal{M}$ is not linear over K. Towards this question, we have the following technical characterization of the cases when $\mathrm{cl}_K^E = \mathrm{scl}_K^E$ holds. We say that a multivariate polynomial is *multilinear* if it has degree one in each variable appearing in it. Let $K \subseteq L$ be a field extension, $E \subseteq L$ a finite subset and $B = \{b_1, \ldots, b_r\} \subseteq E$. We say that E *is multilinear with respect to B (over K)* if B is algebraically independent over K and for each element $x \in E - B$ there are multilinear polynomials $f, g \in K[t_1, \ldots, t_r]$ such that $x = \frac{f(b_1, \ldots, b_r)}{g(b_1, \ldots, b_r)}$.

Theorem 17 *[4, Theorem 8.38] Let K be a field and t a transcendental element over K. If $K(x) = K(t)$ for some $x \in K(t)$, then x has the form*

$$x = \frac{at + b}{ct + d},$$

where $a, b, c, d \in K$, $at + b$ and $ct + d$ are relatively prime in $K[t]$, and $a \neq 0$ or $c \neq 0$ holds.

Theorem 18 *Let K be a field and $K \subseteq L$ a field extension, and let $E \subseteq L$ be a finite subset. Then $\mathrm{cl}_K^E = \mathrm{scl}_K^E$ holds if and only if E is multilinear with respect to B for every basis $B \subseteq E$ of the matroid induced by cl_K^E.*

Proof: Let us write cl for cl_K^E and scl for scl_K^E, and let $\mathcal{M}$ denote the matroid induced by cl. First, suppose that cl = scl. Let $B = \{b_1, \ldots, b_r\} \subseteq E$ be a basis and $x \in E - B$ an element of $\mathcal{M}$ not in this basis. Let C denote the fundamental circuit of x with respect to B in $\mathcal{M}$. Now we have $x \in \mathrm{cl}(C - x) = \mathrm{scl}(C - x)$, so that $x \in K(C - x)$. This means that there are relatively prime polynomials $f, g \in K[t_1, \ldots, t_r]$ in which only the variables corresponding to $C - x$ appear and such that $x = \frac{f(b_1, \ldots, b_r)}{g(b_1, \ldots, b_r)}$. We need to show that f and g are multilinear. Let t_i be a variable appearing in f or g; without loss of generality, we may suppose that $i = r$. Let K' denote $K(t_1, \ldots, t_{r-1})$. Note that $x \in K'(t_r)$, and since $t_r \in \mathrm{cl}(C - t_r) = \mathrm{scl}(C - t_r)$ we also have $t_r \in K'(x)$. Thus we have $K'(x) = K'(t_r)$, and it follows by Theorem 17 that $x = \frac{at_r + b}{ct_r + d}$, where $a, b, c, d \in K'$ and $at_r + b$ and $ct_r + d$ are relatively prime in $K'[t_r]$. Now we have $f \cdot (ct_r + d) = g \cdot (at_r + b)$, which can only happen if $f = w(at_r + b)$ and $g = z(ct_r + d)$ for some elements $w, z \in K'$.[2] This shows that the degree of t_r is at most one in each of f and g. Since t_r was chosen arbitrarily, this shows that f and g are multilinear, as desired.

Now let us suppose that E is multilinear with respect to B for every basis $B \subseteq E$ of $\mathcal{M}$. We first note that cl = scl if and only if for every circuit C of $\mathcal{M}$ and every $x \in C$ we have $x \in \mathrm{scl}(C - x)$. Indeed, let $X \subseteq E$ be an arbitrary subset. Then $x \in \mathrm{cl}(X) - X$ if and only if there is a circuit C with $x \in C \subseteq X + x$. Since we have $\mathrm{scl}(C - x) \subseteq \mathrm{scl}(X)$, it is enough to show that x belongs to the former set. Thus let us consider a circuit C and an element $x \in C$, and let $B = \{b_1, \ldots, b_r\}$ be a basis containing the independent set $C - x$. Now $x \in E - B$, so by assumption there are multilinear polynomials $f, g \in K[t_1, \ldots, t_r]$ such that $x = \frac{f(b_1, \ldots, b_r)}{g(b_1, \ldots, b_r)}$. We may assume that f and g are relatively prime. Let I denote the set of indices i such that t_i appears in f or g. It is enough to show that $\{b_i : i \in I\} \subseteq C - x$, since this implies that $x \in K(C - x)$, so $x \in \mathrm{scl}(C - x)$. Note that if t_i appears in f or g, then we have $x \notin \mathrm{cl}(B - b_i) \supseteq \mathrm{scl}(B - b_i)$. It follows from the mixed exchange property (Lemma 7) that $b_i \in \mathrm{cl}(B + x - b_i)$, and since $b_i \in B$, this holds if and only if $b_i \in C - x$, as desired. $\square$

[2] Here we used the fact that by Gauss' Lemma, f and g are also relatively prime in $K'[t_r]$.

Example 19 *Let K be a field and x, y two indeterminates over K. Consider the set $E = \{x, y, x+y, xy\}$. Clearly, E is multilinear with respect to $\{x, y\}$. The matroid induced by cl_K^E is the uniform matroid $U_{2,4}$. Letting $t = x + y$, we see that $E = \{x, t - x, t, xt - x^2\}$. This shows that E is not multilinear with respect to the basis $B = \{x, x + y\}$. By Theorem 18 this implies that $\mathrm{cl}_K^E \neq \mathrm{scl}_K^E$, and indeed we have*

$$E = \mathrm{cl}_K^E(\{x + y, xy\} \neq \mathrm{scl}_K^E(\{x + y, xy\}) = \{x + y, xy\}.$$

References

[1] M. Beecken, J. Mittmann, N. Saxena, Algebraic independence and blackbox identity testing, *Information and Computation*, **222** (2013)

[2] K. Conrad, Simple radical extensions, *online lecture note*, `https://kconrad.math.uconn.edu/blurbs/galoistheory/simpleradical.pdf`

[3] B. Jackson, J.C. Owen, Equivalent realisations of a rigid graph, *Discrete Applied Mathematics*, **256** (2019)

[4] N. Jacobson, Basic algebra II (2nd ed.), *W.H. Freeman*, New York (1985)

[5] S. Lang, Undergraduate algebra (3rd ed.), *Springer*, New York (2005)

[6] F. Király, L. Theran, R. Tomioka, The Algebraic Combinatorial Approach for Low-Rank Matrix Completion, *Journal of Machine Learning Research*, **16** (2015)

[7] J.G. Oxley, Matroid Theory (2nd ed.), *Oxford University Press*, Oxford, New York (2011)

[8] Z. Rosen, J. Sidman, L. Theran, Algebraic Matroids in Action, *The American Mathematical Monthly*, **127** (2020)

Note on the chromatic number of Minkowski planes: the regular polygon case

Panna Gehér

Department of Operations Research
Eötvös Loránd University
Budapest, Hungary
geherpanni@student.elte.hu

Abstract: The famous Hadwiger–Nelson problem asks for the minimum number of colors needed to color the points of the Euclidean plane so that no two points unit distance apart are assigned the same color. In this note we consider a variant of the problem in Minkowski metric planes, where the unit circle is a regular polygon of even and at most 22 vertices. We present a simple lattice–sublattice coloring scheme that uses 6 colors, proving that the chromatic number of the Minkowski planes above are at most 6. This result is new for regular polygons having more than 8 vertices.

Keywords: Hadwiger–Nelson problem, Colorings of normed planes, Chromatic number, Asymmetric Ramsey-type problems

1 Introduction

In 1950, Nelson raised the following question: What is the minimum number of colors that are needed to color the Euclidean plane so that no two points of the same color determine unit distance? We refer to such a coloring with k color classes as a *proper k-coloring*. Thus Nelson's question asks for the smallest k value such that the plane can be properly k-colored. This value is known as the chromatic number of the Euclidean plane, and is denoted by $\chi(\mathbb{R}^2)$. Immediately after the question was raised the following easy-to-get bounds were established:

$$4 \leq \chi(\mathbb{R}^2) \leq 7.$$

The lower bound is due to Moser [6] who constructed a unit-distance graph (that is a graph whose edges connect vertices unit distance apart) with chromatic number 4. The upper bound is due to Isbell who considered a tilting of the plane by translates of a regular hexagon with diameter slightly less than one and defined a periodic proper 7-coloring.

Despite the numerous attempts to improve these bounds, only little progress was made for more that 60 years – for a historical survey on the problem see [8]. However, in 2018 de Grey [4] constructed a 5-chromatic unit-distance graph, proving that the chromatic number of the plane is at least 5. Shortly afterwards Exoo and Ismailescu [3] independently published another proof.
The problem has regained a lot of attention since the breakthrough and a Polymath project was launched with the main goal of creating a human-verifiable proof of the new result. Although the proofs are still relying on computers, quite some progress has been made: while the distance graph published by de Grey had a total of 1581 vertices, the current known smallest example consists only 509 [7]. As a consequence of the breakthrough many variations of the problem have gained more attention in the last couple of years. One interesting research area is generalizing the question to Minkowski planes: Let C be a two-dimensional centrally symmetric bounded convex domain centered at the origin and let $(\mathbb{R}^2, C)$ denote the Minkowski plane where C determines the unit circle; the C-norm of an $x \in \mathbb{R}^2$ point is the value:

$$\|x\|_C := \min\left\{\lambda \in \mathbb{R}^+ | x \in \lambda C\right\}.$$

The C-distance of two points x and y is defined by $\|x-y\|_C$. Naturally, the chromatic number of the Minkowski planes – denoted by $\chi(\mathbb{R}^2,\ C)$ – is the minimum number of colors needed to color the points of $\mathbb{R}^2$ such that no monochromatic point pair determines a unit C-distance. The main result concerning the chromatic number of Minkowski planes is due to Chilakamarri [1]: by extending the arguments of Moser and the construction of Isbell he proved that the bounds

$$4 \leq \chi(\mathbb{R}^2,\ C) \leq 7$$

hold for all centrally symmetric bounded convex domain C. An interesting special case of the above problem is when the unit circle is a regular polygon of even number of vertices. Or more generally we can consider any affine image of a regular polygon since the problem itself is affine invariant. The study of the chromatic number of such normed planes was also initiated by Chilakamarri who considered the cases of regular polygons with few vertices. He gave a tile-based proper 4-coloring for the parallelogram and the symmetric hexagon's case, proving that the answer here is exactly 4. He also gave a proper 6-coloring in case C is a centrally symmetric octagon: he considered a packing of $C/2$, that is a packing of circles of radius half. He showed that the translates of $C/2$ can be colored using 4 colors and two more colors can take care of the remaining squares (for details see Figure 1). It is worth noting that we have to be careful with choosing the colors of boundary points of the octagons as no antipodal point pair can have the same color.

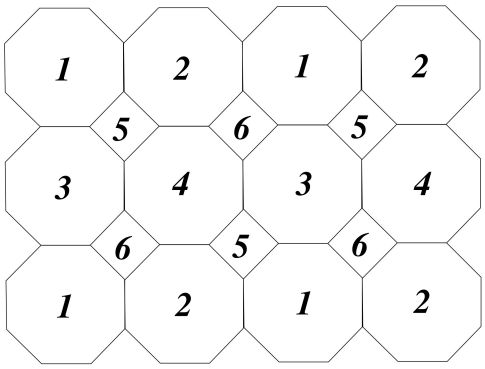


Figure 1: Chilakamarri's proper 6-coloring of the Minkowski plane equipped with the regular octagon metric

Chilakamarri asked whether or not the chromatic number of a plane equipped with the norm defined by the regular octagon or any centrally symmetric octagon is exactly 4. No progress was made until very recently when Exoo, Fisher and Ismailescu [2] answered his question negatively: they constructed 5-chromatic unit-distance graphs in the cases of regular polygons with 8, 10, and 12 vertices. Together with the Euclidean case these are the only known examples of a normed plane with chromatic number at least 5. Table 1. summarizes the mentioned results:

Unit circle C	$\chi(\mathbb{R}^2,\ C)$
Parallelogram, centrally symmetric hexagon (see [1])	4
Regular octagon (see [2] and [1])	5 or 6
Regular decagon, regular dodecagon (see [2] and [1])	5, 6 or 7
Euclidean circle (see [4], [3] and [1])	5, 6 or 7
Arbitrary symmetric convex domain (see [1])	4, 5, 6 or 7

Table 1: Possible values of the chromatic numbers of Minkowski planes

2 Main result

In this note we extend Chilakamarri's result for regular octagons to regular polygons with at most 22 vertices by giving a simple lattice–sublattice coloring scheme that uses only 6 colors. It also slightly strengthens the result of Chilakamarri as our colorings are regular: We call a proper k-coloring of $\mathbb{R}^2$ with color classes $S_1, S_2 \dots S_k$ *regular*, if there exist vectors $v_1 \dots, v_k$ such that $S_i = S_1 + v_i$ for $i = 1 \dots k$, that is the color classes are translates of each other. Now we state our main theorem:

Theorem 1. *Let C be a regular polygon with an even number of vertices. In case C has at most 22 vertices then there exists a regular proper 6-coloring of the Minkowski plane equipped with the C-metric such that no points unit C-distance apart are identically colored. Hence, if C is a regular $2k$-gons, where $k \leq 11$, then:*

$$\chi(\mathbb{R}^2,\ C) \leq 6.$$

2.1 The coloring scheme

Let C be a regular octagon first and define a symmetric convex hexagon H inscribed in $C/2$ as follows: Choose two opposite sides of $C/2$ and form a hexagon using their four endpoints and two additional boundary points of $C/2$. The choice of the additional points can be made in various ways, here we simply chose the ones that halve the boundary line of $C/2$ connecting the chosen sides. Denote the vertices of H by A_i $(i = 1 \dots 6)$ in a clockwise order as shown in Figure 2.

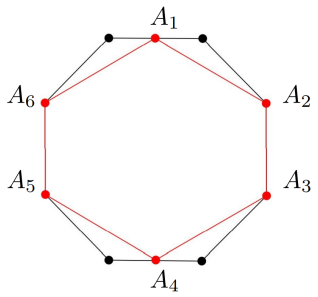


Figure 2: The centrally symmetric hexagons inscribed in $C/2$ in the case of the regular decagon

To avoid unit C-distance in H, remove the boundary points lying between the points A_1 and A_4, including A_4 but not A_1. In this way no antipodal point pair is monochromatic. For simplicity call the resulting half-open hexagon still H. Now consider a tiling of the plane by translates of H and assign colors 1 through 6 periodically as shown in Figure 3. We can assume that the centers of the hexagons form a lattice, that we denote by $\mathcal{L}$.

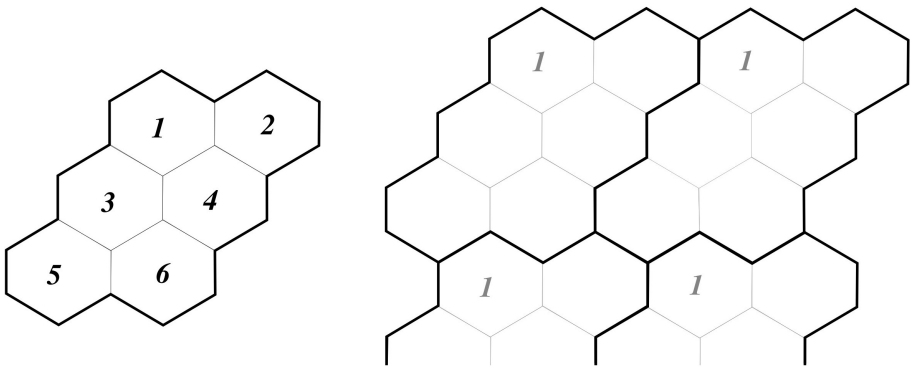


Figure 3: A tiling of the plane with translates of hexagon H defines a periodic 6-coloring

Note that the main difference between this coloring scheme and Chilakamarri's general 7-coloring is that here we can take advantage of C not being strictly convex: in the direction perpendicular to the sides shared with $C/2$ the monochromatic hexagons can be placed such that they are separated by only one differently colored hexagon.

Now we justify that our coloring is proper: as mentioned earlier unit C-distance is not realized within the hexagons. All is left is to check that two hexagons of the same color are too far from each other to determine a point pair unit C-distance apart. As the color classes are congruent, it is enough to verify the statement for one specific color class, say the class of red points. We can also assume that one of the red hexagons has the origin as its center, thus the set of centers of red hexagons form a sublattice $\mathcal{L}'$. By the symmetry of C it is enough to show that polygons $\mathcal{L}' + C/2 \oplus H$ form a packing, where $\oplus$ denotes the Minkowski sum of the two polygons, that is:

$$C/2 \oplus H = \{c + h \mid c \in C/2,\ h \in H\}.$$

Straightforward calculations finish the proof: Without loss of generality we can assume that C has circumradius one. Let v_1 and v_2 denote the basis vectors of the lattice $\mathcal{L}'$ where we can assume that v_1 is perpendicular to the sides shared with $C/2$. Then for any vector $\lambda \in \mathcal{L}'$, polygons $\lambda + C/2 \oplus H$ and $\lambda \pm v_1 + C/2 \oplus H$ are trivially disjoint. From definition $H + C/2 \subseteq C$ so $H + C/2$ also has circumradius at most one. Hence it is enough to check that with the exception of $\underline{0}$ and $\pm v_1$ any lattice vector has Euclidean length at least two. This obviously holds (see Figure 4), thus the coloring is indeed proper.

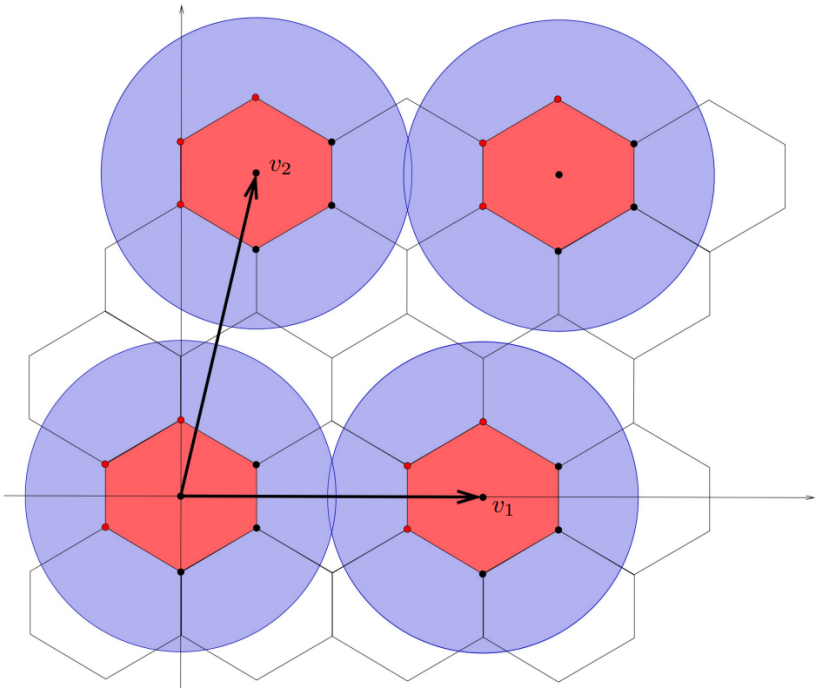


Figure 4: For a vector v of $\mathcal{L}' \setminus \{\pm v_1, \underline{0}\}$ the Euclidean unit circle B_2 is disjoint from all translates $B_2 + v$

Now consider regular polygons with greater number of vertices. We show that almost the same coloring scheme works for all the remaining cases: Two sides of H can always be two opposite sides of $C/2$, we only have to be careful with the choice of the remaining two vertices. For the case of a regular 10- and 12-gons choosing the halving points on the boundary line of $C/2$ still works, we can simply define hexagon H as shown in Figure 5.

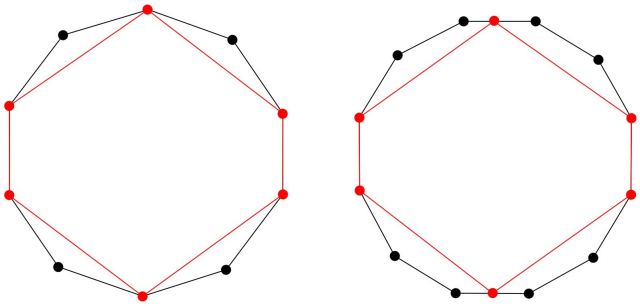


Figure 5: The centrally symmetric hexagons inscribed in $C/2$ chosen in the case of the regular 10- and 12-gon

However, in the remaining cases we had to flatten the hexagons in order to get a proper coloring. In the case of regular 14-, 16- and 18-gons some other vertices of $C/2$ were chosen. But in the final two cases only non-vertex points seemed to be working: for $n = 20$ bisectors of some other sides were chosen, and for $n = 22$ we divided one of the sides in the ratio $0.68 : 0.32$. For details see Figure 6.

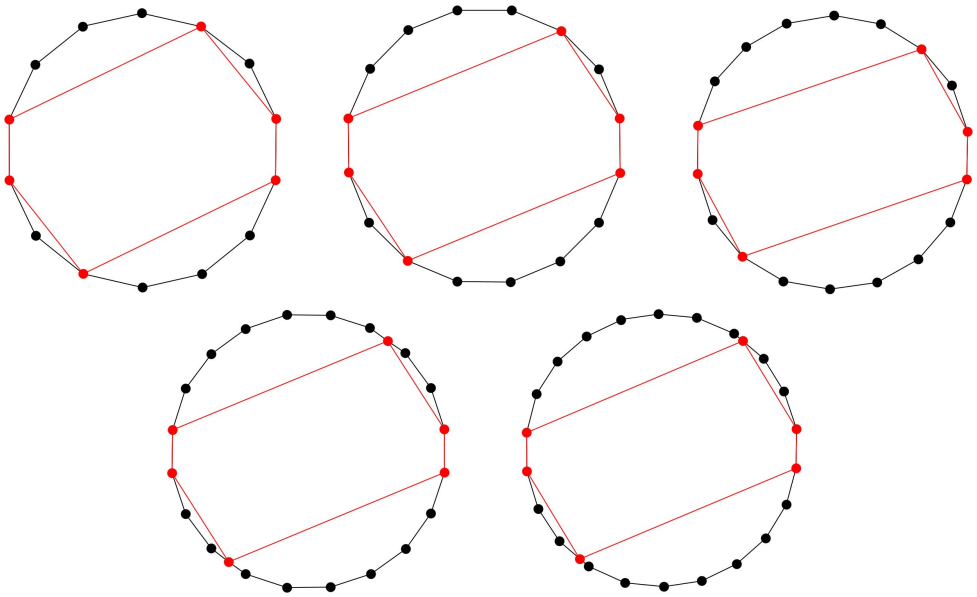


Figure 6: The symmetric hexagon H inscribed in $C/2$ chosen for the regular 14-, 16-, 18-, 20- and 22-gon

2.2 The regular dodecagon's case

Now we present the details of the proof in the case of the regular dodecagon. Consider the regular dodecagon centered at the origin with circumradius 2, whose vertices are:

$$\big(\pm 1, \pm\sqrt{3}\big), \big(\pm\sqrt{3}, \pm 1\big), \big(\pm 2, 0\big), \big(0, \pm 2\big).$$

Let H be the symmetric hexagon inscribed in $C/2$ as defined in Section 2.1: take two opposite sides of $C/2$, for example the sides parallel to vector $(2 - \sqrt{3}, 1)$ and choose the two additional points such that they halve parts of the boundary line of $C/2$ between the chosen sides. Denote these six vertices by A_i $(i = 1 \dots 6)$ in a clockwise order as shown in Figure 7.

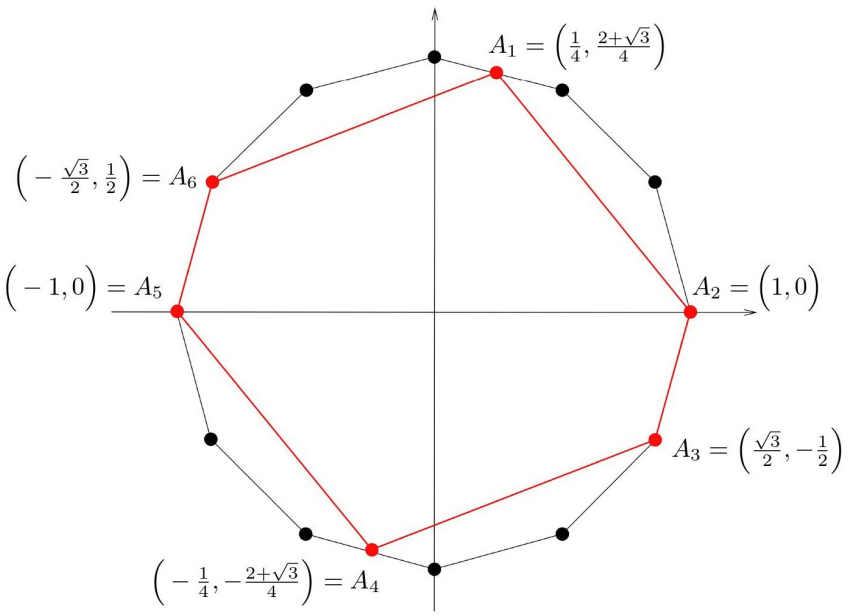


Figure 7: H hexagon inscribed in $C/2$

As before, let H be the half-open hexagon defined by the points A_i that does not contain the line segment connecting the points A_1 and A_4 and the point A_1 itself. Therefore the hexagonal tiling of the plane with hexagon H is the packing by Voronoi regions of the lattice $\mathcal{L}$ spanned by vectors $\left(\frac{1-2\cdot\sqrt{3}}{4}, \frac{4+\sqrt{3}}{4}\right)$ and $\left(\frac{2+\sqrt{3}}{2}, -\frac{1}{2}\right)$. The basis vectors of the sublattice $\mathcal{L}'$ corresponding to the single color class containing the hexagon centered at the origin are:

- $v_1 = \left(\frac{3-6\cdot\sqrt{3}}{4}, \frac{12+3\cdot\sqrt{3}}{4}\right)$,
- $v_2 = \left(2+\sqrt{3}, -1\right)$.

As mentioned in Section 2.1 what we need to show is that polygons $\mathcal{L}' + C/2 \oplus H$ form a packing. The vertices of $C/2 \oplus H$ are:

$$B_1 = \left(\frac{1}{4}, \frac{6+\sqrt{3}}{4}\right),\ B_2 = \left(\frac{3}{4}, \frac{2+3\cdot\sqrt{3}}{4}\right),\ B_3 = \left(\frac{1+2\cdot\sqrt{3}}{4}, \frac{4+\sqrt{3}}{4}\right),\ B_4 = \left(\frac{2+\sqrt{3}}{2}, \frac{1}{2}\right),$$

$$B_5 = \left(2, 0\right),\ B_6 = \left(\sqrt{3}, -1\right),\ B_7 = \left(\frac{1+\sqrt{3}}{2}, -\frac{1+\sqrt{3}}{2}\right),\ B_8 = \left(\frac{1}{4}, -\frac{2+3\cdot\sqrt{3}}{4}\right) \text{ etc.}$$

The coordinates of the remaining vertices can be obtained by symmetry.

As the coloring is regular, it is enough to pick hexagon H centered at the origin and show that $H \oplus 1/2C$ is disjoint from $\lambda' + H \oplus 1/2C$ for all $\lambda' \not\equiv 0$ in $\mathcal{L}'$.

By definition $H \oplus C/2$ has circumradius 2. Inside the circle of radius 2 centered at the origin there are 4 lattice points of $\mathcal{L}'$ besides the origin and by symmetry we only have to check 2 of them and the corresponding hexagons, namely:

- $H_1 := H + v_2$ and
- $H_2 := H + v_1 + v_2$.

H and H_1 are separated by exactly one differently colored hexagon which is enough as v_2 is perpendicular to the common sides of H and $C/2$. All is left is to give a line that separates H from H_2. For example consider the line l defined by equation:

$$y = -\frac{2+\sqrt{3}}{3}x + \frac{5+2\cdot\sqrt{3}}{3}.$$

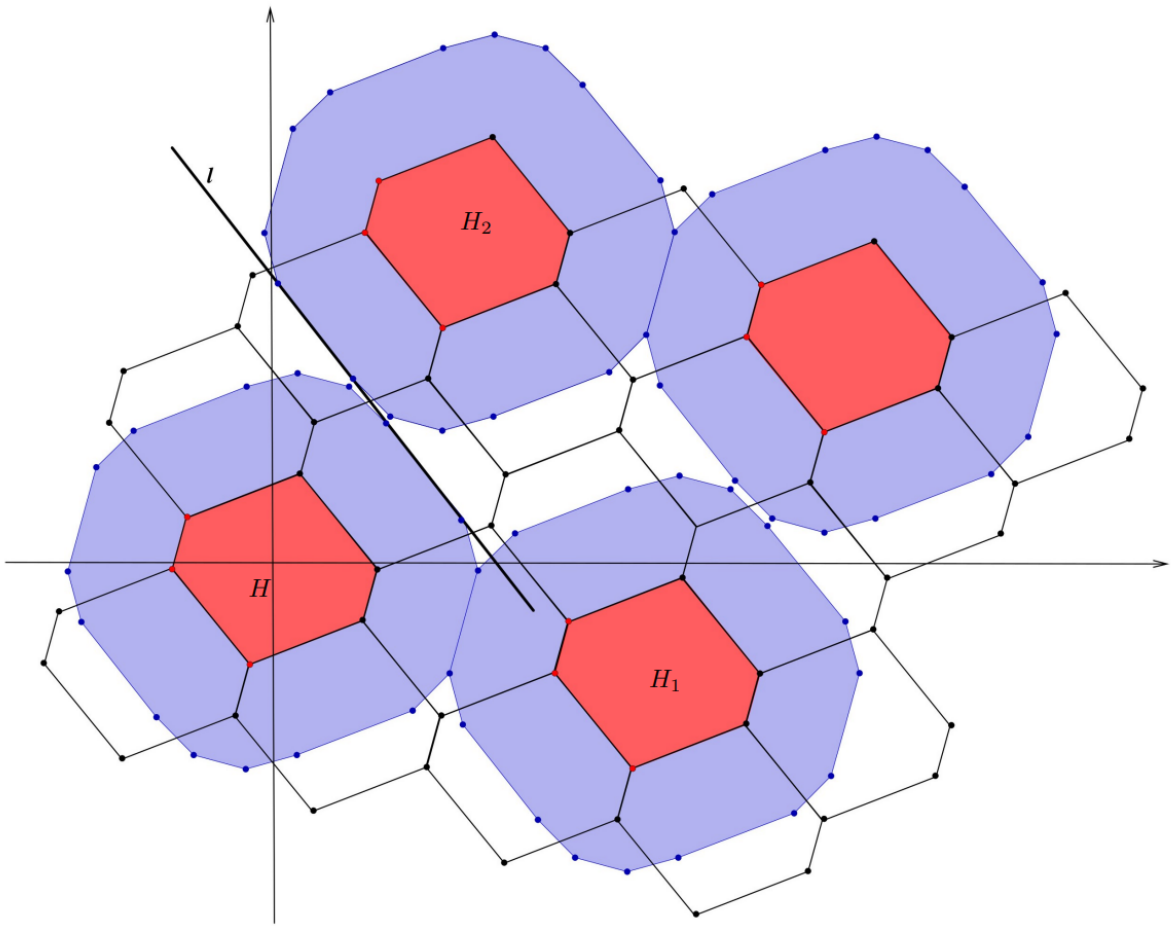


Figure 8: Line l separates $H \oplus C/2$ and $H_2 \oplus C/2$

It is straightforward to check that l goes through two parallel sides of $H \oplus C/2$ and $H_2 \oplus C/2$ (which are on the opposite sides of their centers) and the remaining vertex points of $H \oplus C/2$ are below line l, while all of the remaining vertex points of $H_2 \oplus C/2$ are above it (see Figure 8). Therefore $H \oplus C/2$ and $H_2 \oplus C/2$ are disjoint, thus the coloring is proper.

We remark that in the presented example one can define hexagon H in many different ways as the coloring scheme is quite flexible in this case. However, as the number of vertices increases, the range of possible choices narrows down quickly. For example in the case of the regular 22-gon we have to be really carefull with the definition of hexagon H: as Figure 9 shows, our coloring is almost rigid.

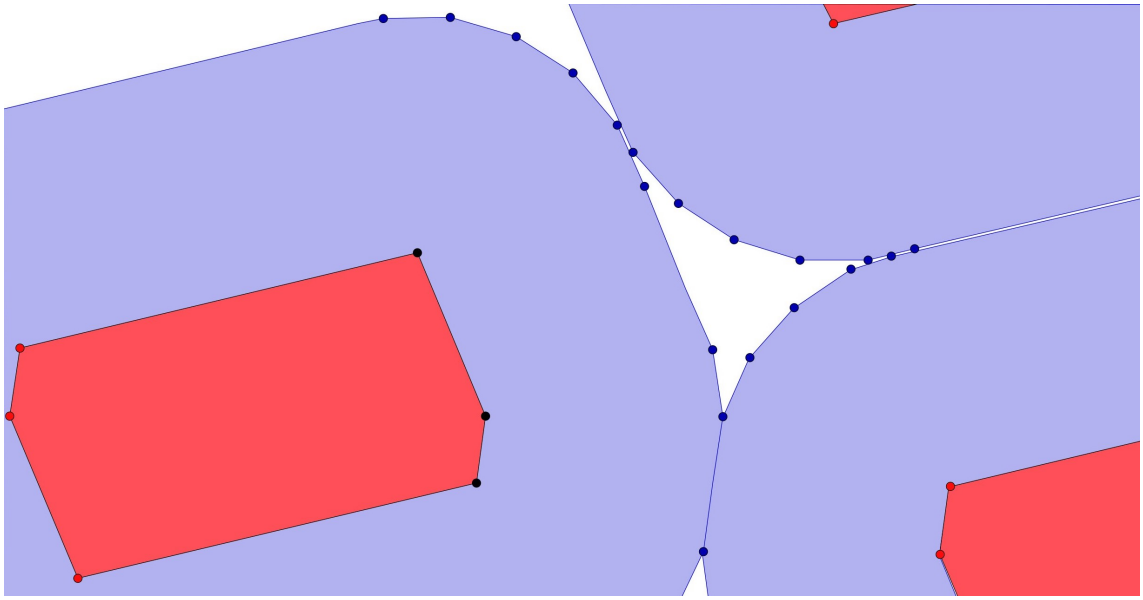


Figure 9: In the 6-coloring of the Minkowski plane equipped with the regular 22-gon metric monochromatic hexagons get dangerously close together

3 An application: an asymmetric Ramsey-type problem

Another direction for generalizing the Hadwiger–Nelson problem is to replace the pair of points at unit distance by another finite point configuration. Moreover we can look for different configurations in each color class. In the rest of the paper we are interested in the following question: for a given $(\mathbb{R}^2,\ C)$ Minkowski plane and a given point configuration $K \subseteq \mathbb{R}^2$ is it true that in any red-blue coloring of $\mathbb{R}^2$ there are either two red points C-distance one apart or there is a blue translate of configuration K? Let $k(C)$ denote the largest value k such that the answer is 'yes' for all configuration K of at most k points. This problem was first considered in [9, 5] by Szlam and Johnson who showed the chromatic number of $(\mathbb{R}^2,\ C)$ provides a lower bound on the value of $k(C)$. For the sake of completeness, we include their short proof as well.

Lemma 2 (Szlam and Johnson [5])**.** *Let $(\mathbb{R}^2,\ C)$ be a Minkowski plane and assume that there exists a k-point configuration K and a red-blue coloring of $\mathbb{R}^2$ such that the red color class avoids unit C-distance and the blue color class avoids all translates of K. Then $\mathbb{R}^2$ can be properly k-colored that is none of the k color classes contains unit C-distance. Hence $k(C) + 1 \geq \chi(\mathbb{R}^2)$.*

Proof: *Assume that we are given a configuration $K = \{a_1, \dots a_k\}$ and a red-blue coloring of $\mathbb{R}^2$ with the desired properties. Then, for each $x \in \mathbb{R}^2$ there is at least one index i such that $x + a_i$ is red. Now let us color the point x with color number i: as there are no red points unit C-distance apart, this indeed defines a proper k-coloring.* □

Szlam also a gave a partial converse to Lemma 2 that considers only regular colorings:

Lemma 3 (Szlam [9])**.** *Assume that $(\mathbb{R}^2,\ C)$ can be properly k-colored by a regular coloring, with color classes $S_i = S_1 + v_i$ (for $i = 1 \dots k$). Then there exists a red-blue coloring of $\mathbb{R}^2$ and a k-point configuration, namely $K = \{v_1, v_2, \dots v_k\}$ such that the red color class avoids unit C-distance and the blue color class avoids all translates of K.*

As the 6-colorings described in Theorem 1 are a regular, we immediately get the following corollary:

Corollary 4. *Let* $(\mathbb{R}^2;\ C)$ *be a Minkowski plane whose unit circle is a regular polygon with an even number of vertices. In case* C *has at most* 22 *vertices then there exists a red-blue coloring of the plane and a configuration* K *of* 6 *points such that there is no red point pair unit* C*-distance apart, and the blue color class avoids all translates of* K*.*

We finish the paper with a small remark on Szlam's results: We noticed that although the proof of Lemma 3 is short and straightforward, it can be a bit misleading. To see its inconvenience notice how the proper coloring defined in Lemma 2 is not regular: color classes are generated by covering the plane with translates of the unit distance avoiding red set. Call a coloring with such structure subregular. More precisely we call a proper k-coloring with color classes $S_1 \ldots S_k$ *subregular* if there exist vectors $v_1 \ldots v_k$ such that S_i is a subset of $S_1 + v_i$. We show that Lemma 3 can be extended to subregular colorings in a very natural way:

Theorem 5. *Let* $(\mathbb{R}^2;\ C)$ *be a given Minkowski plane. Assume that* $\mathbb{R}^2$ *can properly be* k*-colored by a subregular coloring defined by a* C*-unit distance avoiding set* S_1 *and vectors* $v_1,\ v_2 \ldots v_k$*. Then there exists a red-blue coloring of* $\mathbb{R}^2$ *and a* k*-point configuration, namely* $K = \{-v_1, -v_2, \cdots - v_k\}$ *such that the red color class avoids unit* C*-distance and the blue color class avoids all translates of* K*.*

Proof: *Let the points of* S_1 *be colored red, and color all the remaining points blue. As promised, let us consider the configuration* $K = \{-v_1, -v_2, \ldots, -v_k\}$*. We wish to show that for an arbitrary vector* m *color class* S_1 *contains at least one point of* $K + m$*. Without loss of generality we can assume* $v_1 \equiv 0$*. Hence if* $m \in S_1$ *there is nothing to prove. Assume that* $m \notin S_1$*. In this case there exists an index* i *such that* $m \in S_1 + v_i$ *which leads to* $-v_i + m \in S_1$*.* $\square$

Finally, we note that analogous results considering higher dimensional Minkowski spaces can be achieved. Let $k_n(C)$ denote the largest k value such that in any red-blue coloring of the Minkowski space determined by $C \subseteq \mathbb{R}^n$ there are either two red points unit C-distance apart or there is a blue translate of any configuration with at most k points. An easy observation is that both Lemma 2 and Theorem 5 can be extended to Minkowski spaces and it follows that for all n and $C \subseteq \mathbb{R}^n$ the value $k_n(C)$ is exactly the smallest number k such that there exists a subregular k-coloring of $(\mathbb{R}^n;\ C)$.

References

[1] K. B. Chilakamarri, Unit-distance graphs in Minkowski metric spaces, *Geometriae Dedicata* **345-356** (1991)

[2] G. Exoo, D. Fisher, D. Ismailescu, The chromatic number of the Minkowski plane – the regular polygon case, *arXiv preprint* **arXiv:2108.12861** (2021)

[3] G. Exoo, D. Ismailescu, The chromatic number of the plane is at least 5: A new proof, *Discrete & Computational Geometry* **64(1), 216-226** (2020)

[4] A. de Grey, The Chromatic Number of the Plane Is at least 5, *arXiv preprint* **arXiv:1804.02385** (2018)

[5] P. D. Johnson, A. D. Szlam, A New Connection Between Two Kinds of Euclidean Coloring, *Geombinatorics* **10(4), 172-178** (2001)

[6] L. Moser, W. Moser, Solution to problem 10, *Can. Math. Bull.* **4. 187–189** (1961)

[7] J. Parts, Graph minimization, focusing on the example of 5-chromatic unitdistance graphs in the plane, *arXiv preprint* **arXiv:2010.12665** (2020)

[8] A. Soifer, The mathematical coloring book *New York: Springer* (2008)

[9] A. D. Szlam , Monochromatic translates of configurations in the plane, *Journal of Combinatorial Theory, Series A* **93(1), 173-176.** (2001)

Widely colorable graphs and their multichromatic numbers

Anna Gujgiczer[1]

Department of Computer Science and Information Theory
Budapest University of Technology and Economics
and
MTA-BME Lendület Arithmetic Combinatorics Research Group, ELKH, Budapest, Hungary
gujgicza@cs.bme.hu

Gábor Simonyi[2]

Alfréd Rényi Institute of Mathematics, Budapest, Hungary
and
Department of Computer Science and Information Theory
Budapest University of Technology and Economics
simonyi@renyi.hu

Abstract: Answering a question of Claude Tardif we show that if a graph admits a so-called s-wide coloring using t colors then its s-fold chromatic number is at most $t + 2(s - 1)$. The talk is based on the paper [2].

Keywords: homomorphism, Kneser graphs, multichromatic number, wide coloring

1 Introduction

For every pair of positive integers n, k satisfying $n \geq 2k$ the Kneser graph $\mathrm{KG}(n, k)$ is defined in the following way. Its vertices are the $\binom{n}{k}$ k-element subsets of $[n] = \{1, \dots n\}$ and its edges are formed by pairs of disjoint subsets. The study of multichromatic numbers goes back to Stahl [8] whose conjecture about the multichromatic numbers of Kneser graphs (that can also be expressed by the existence and non-existence of graph homomorphisms between different Kneser graphs) is still wide open, see the book [5] for further information. In short we can say that the k-fold chromatic number $\chi_k(G)$ of a graph G is the smallest n for which using n colors in total we can assign k distinct colors to each vertex of G in a way that no color appears on adjacent vertices. It is easy to see that this is equivalent to say that n is the smallest positive integer for which G admits a homomorphism to the Kneser graph $\mathrm{KG}(n, k)$.

Wide colorings provide another graph coloring concept that turned out to be relevant in several contexts. For every $s \geq 1$ an s-wide coloring of a graph G is a coloring of its vertices in such a way that no walk of length $2s - 1$ can start and end in the same color class. In particular, a 1-wide coloring is just a proper coloring, a 2-wide coloring is a proper coloring with the additional property that the (first) neighborhood of any color class is also an independent set. In general, an s-wide coloring is a proper coloring where the first, second, $\dots, (s - 1)^{\text{th}}$ neighborhood of any color class is also an independent set. It is obvious that if a graph G admits an s-wide coloring then its odd girth (the length of its shortest odd cycle) $g_o(G)$ should be at least $2s + 1$. On the other hand, if $g_o(G) \geq 2s + 1$ then a coloring assigning a different color to every vertex is certainly s-wide. The concept becomes more interesting if we do not need to use more colors for an s-wide coloring than for any proper coloring. In fact, it was a question of Harvey and Murty whether there exist t-chromatic graphs that admit a t-coloring which is 2-wide. This was answered affirmatively by Gyárfás, Jensen and Stiebitz in [3]. 3-wide colorings turned out to be relevant

[1]Research is partially supported by the National Research, Development and Innovation Office (NKFIH) grant K–120706 of NKFIH Hungary.

[2]Research is partially supported by the National Research, Development and Innovation Office (NKFIH) grants K–120706, K–132696 and SNN-135643 of NKFIH Hungary.

in connection to investigations of the local chromatic number, see [7]. More recently, s-wide colorability was also used in the context of finding counterexamples to Hedetniemi's conjecture with small chromatic number, cf. [12, 9, 13, 10].

In the talk, which is based on the paper [2], a result about the multichromatic numbers of s-wide-colorable graphs will be presented that answers a question asked by Claude Tardif in [9].

2 The graphs $W(s,t)$ and their multichromatic numbers

It can be shown that a graph G admits an s-wide coloring with t colors if and only if there exists a homomorphism from G into a certain universal graph we denote by $W(s,t)$. These graphs came up in different forms in the papers [3, 1, 7, 4, 12]. One of their possible definitions is as follows.

Definition 1

$$V(W(s,t)) = \{(x_1 \dots x_t) : \ \forall i \ x_i \in \{0,1,\dots,s\}, \exists! i \ x_i = 0, \ \exists j \ x_j = 1\},$$

$$E(W(s,t)) = \{\{(x_1 \dots x_t),(y_1 \dots y_t)\} : \forall i \ |x_i - y_i| = 1 \text{ or } x_i = y_i = s\}.$$

Using the topological method introduced by Lovász in his celebrated work [6] on Kneser graphs it is shown in the above mentioned papers that $\chi_1(W(s,t)) = \chi(W(s,t)) = t$ for all meaningful values of the parameters s and t.

The motivation for our work came from Tardif [9] who observed that $\chi_2(W(s,t)) = t+2$ when $s = 2$ and in general

$$\chi_k(W(s,t)) \geq t + 2(k-1)$$

holds for every k. He asked whether we will have strict inequality for $k = s = 3$. Our main result answers this in the negative, in fact, we proved the following more general theorem.

Theorem 2 ([2]) *If $k \leq s$, then*

$$\chi_k(W(s,t)) = t + 2(k-1).$$

Nevertheless, asymptotically Tardif's guess was correct as one can also prove (as he also noted [11]) that the following holds.

Proposition 3 *For all pairs of positive integers $t \geq 3$ and $s \geq 1$ there exists some threshold $k_0 = k_0(s,t) > s$ for which*

$$\chi_k(W(s,t)) > t + 2(k-1)$$

whenever $k \geq k_0$.

It would be interesting to know whether the smallest possible k_0 in Proposition 3 is $s+1$, as our result may suggest, or larger.

Finally, we remark that since a graph G admits an s-wide coloring with t colors if and only if there exists a homomorphism from G to $W(s,t)$, and a graph F has $\chi_k(F) \leq n$ if and only if it admits a homomorphism to the Kneser graph $\mathrm{KG}(n,k)$, Theorem 2 implies that if G is a graph that admits an s-wide coloring with t colors, then

$$\chi_k(G) \leq t + 2(k-1)$$

whenever $k \leq s$.

References

[1] Stephan Baum, Michael Stiebitz, Coloring of graphs without short odd paths between vertices of the same color class, unpublished manuscript, 2005.

[2] Anna Gujgiczer, Gábor Simonyi, On multichromatic numbers of widely colorable graphs, *J. Graph Theory* **100** (2022), 346-361.

[3] András Gyárfás, Tommy Jensen, Michael Stiebitz, On graphs with strongly independent color-classes, *J. Graph Theory* **46** (2004), 1–14.

[4] Hossein Hajiabolhassan, On colorings of graph powers, *Discrete Math.* **309** (2009), 4299–4305.

[5] Pavol Hell, Jaroslav Nešetřil, *Graphs and Homomorphisms*, Oxford University Press, New York, 2004.

[6] László Lovász, Kneser's conjecture, chromatic number, and homotopy, *J. Combin. Theory, Ser. A* **25** (1978), 319–324.

[7] Gábor Simonyi and Gábor Tardos, Local chromatic number, Ky Fan's theorem, and circular colorings, *Combinatorica* **26** (2006), 587–626.

[8] Saul Stahl, n-Tuple colorings and associated graphs, *J. Combin. Theory, Ser. B* **20** (1976), 185–203.

[9] Claude Tardif, The chromatic number of the product of 14-chromatic graphs can be 13, *Combinatorica* **42** (2022), 301–308.

[10] Claude Tardif, The chromatic number of the product of 5-chromatic graphs can be 4, *manuscript*, available at `https://www.researchgate.net/publication/365650263_THE_CHROMATIC_NUMBER_OF_THE_PRODUCT_OF_5-CHROMATIC_GRAPHS_CAN_BE_4` , 2022.

[11] Claude Tardif, private communication, 2020.

[12] Marcin Wrochna, On inverse powers of graphs and topological implications of Hedetniemi's conjecture, *J. Combin. Theory, Ser. B* **139** (2019), 267–295.

[13] Marcin Wrochna, Smaller counterexamples to Hedetniemi's conjecture, arXiv:2012.13558 [math.CO], 2020.

Abstract Rigidity Matroids of Uniform Hypergraphs

Mizuki Higashida

Department of Mathematical Informatics
University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo, Japan
higashida-mizuki405@g.ecc.u-tokyo.ac.jp

Shin-ichi Tanigawa[1]

Department of Mathematical Informatics
University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo, Japan
tanigawa@mist.i.u-tokyo.ac.jp

Abstract: J. Graver introduced abstract rigidity matroids of graphs by axiomizing the gluing property of graph rigidity. In this paper we extend Graver's idea to uniform hypergraphs. We show that the class of abstract rigidity matroids of hypergraphs captures the combinatorics behind generalized stresses of simplicial complexes and cofactors of multivariate splines.

Keywords: rigidity matroid, graph rigidity, splines, hypergraphs, generalized stresses

1 Introduction

The central open problem in graph rigidity theory is to prove a combinatorial characterization of graphs that are generically rigid in d-space. This problem is equivalently formulated as giving a combinatorial rank formula of the d-dimensional generic rigidity matroid. To gain a better understanding of this hard question, J. Graver [6] proposed a matroidal approach which analyzes a new class of matroids defined by axiomizing representative properties of rigid graphs. Specifically, Graver focused on the following well-known gluing property of rigid graphs:

- The union of two rigid graphs G_1 and G_2 is rigid if they share at least d vertices.
- The union of two graphs G_1 and G_2 is not rigid if they share at most $d-1$ vertices. Moreover, the internal degree of freedom of each G_i does not change in the union.

In terms of rigidity matroids, these two gluing properties can be written as

(R1) If $E_1, E_2 \subseteq K(V)$ with $\mathrm{cl}_{\mathcal{M}}(E_1) = K(V(E_1)), \mathrm{cl}_{\mathcal{M}}(E_2) = K(V(E_2))$, and $|V(E_1) \cap V(E_2)| \geq d$, then $\mathrm{cl}_{\mathcal{M}}(E_1 \cup E_2) = K(V(E_1 \cup E_2))$, and

(R2) If $E_1, E_2 \subseteq K(V)$ with $|V(E_1) \cap V(E_2)| \leq d-1$, then $\mathrm{cl}_{\mathcal{M}}(E_1 \cup E_2) \subseteq K(V(E_1)) \cup K(V(E_2))$,

where $\mathrm{cl}_{\mathcal{M}}$ denotes the closure operator of the underlying matroid $\mathcal{M}$, $K(V)$ denotes the edge set of the complete graph of a finite set V, and $V(E)$ denotes the set of vertices spanned by E for each $E \subseteq K(V)$. Since (R1) and (R2) are written without a reference to graph rigidity (written only in terms of the underlying graph and matroid), Graver considered (R1) and (R2) as a new axiom for defining a class of matroids. Formally, a matroid $\mathcal{M}$ on the edge set of a complete graph is said to be an *abstract d-rigidity matroid* if (R1) and (R2) hold in $\mathcal{M}$.

The d-dimensional generic rigidity matroid (of a complete graph) is a primary example of abstract d-rigidity matroids. Later Whiteley [16] found a new substantial example of abstract rigidity matroids from approximation theory. This matroid, known as the *cofactor matroid*, characterizes the space of cofactors in the spline spaces over polyhedral domains. A recent result by Clinch, Jackson, and Tanigawa [3, 4]

[1]Research is supported by JST PRESTO Grant Number JPMJPR2126 and JSPS KAKENHI Grant Number 18K11155.

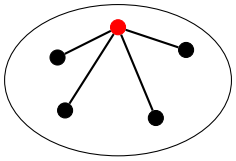


Figure 1: A 4-valent extension of a graph.

reveals a significance of Graver's approach by exhibiting a solution to the cofactor counterpart of the 3-dimensional rigidity problem.

In order to understand and extend Stanley's g-theorem [12] for a characterization of the f-vectors of simplicial polyhedrons, C. Lee [8, 9] introduced the concept of *generalized stresses*. This is an extension of self-stresses in graph rigidity to simplicial complexes, and has been extensively studied in the context of polyhedral combinatorics, see, e.g. [1, 11] for recent papers. A graph rigidity view of generalized stresses has been developed by Tay, White, and Whiteley [13, 14, 16], where the *skeletal rigidity* and the underlying *skeletal rigidity matroids* of simplicial complexes were defined.

The corresponding theory for spline cofactors is also known. This was implicit in the homological approach of Billera [2] for the analysis of multivariate splines over simplicial complexes, and later Whiteley [16] made this connection explicit by introducing the cofactor matroids of simplicial complexes.

In view of these parallel theories of skeletal rigidity and splines in simplicial complexes, it is a natural research direction to extend Graver's idea of abstract rigidity to hypergraphs. Whiteley [16] however pointed out that giving a proper extension of abstract rigidity is challenging because there is no natural extension of (R1)(R2) in the skeletal rigidity matroid. The main contribution of this work is to give the right notion of abstract rigidity of uniform hypergraphs. The key ingredient in our result is Nguyen's characterization of abstract rigidity matroids that gives several different ways to define abstract rigidity matroids [10]. We show that all the properties in Nguyen's characterization, except the gluing property, can be extended to hypergraphs, and thus abstract rigidity matroids can be defined based on any one of those equivalent properties. We further show that the skeletal rigidity matroid and the cofactor matroid are indeed included in the class of abstract rigidity matroids. We then look at the hypergraph extension of the maximality conjecture of abstract rigidity matroids, and observe an interesting difference between hypergraphs and ordinary graphs.

2 Abstract Rigidity Matroids

Our goal in this section is to define abstract rigidity matroids by extending Nguyen's result. One property used in Nguyen's characterization is the extension property defined by a graph operation. For a graph G, a *d-valent extension* is an operation that creates a new graph from G by adding a new vertex with d distinct new edges incident to the new vertex. See Figure 1.

Graver, Servatius, and Servatius [7] initiated an investigation of alternative definitions of abstract rigidity matroids. Bulding on their result, Nguyen [10] gave the following list.

Theorem 1 (Nguyen [10]) *Let n, d be positive integers with $n \geq d+2$ and $\mathcal{M}$ be a matroid on the edge set of the complete graph with n vertices. Consider the following four properties:*

(P1) the rank of $\mathcal{M}$ is $dn - \binom{d+1}{2}$;

(P2) the edge set of each subgraph isomorphic to K_{d+2} is a circuit;

(P3) the edge set of each subgraph isomorphic to $K_{1,n-1}$ minus $d-1$ edges is a cocircuit;

(P4) each d-valent extension preserves the independence of the edge sets of graphs.

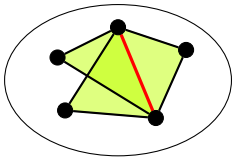


Figure 2: A 3-valent extension of a 3-uniform hypergraph.

Then

$\mathcal{M}$ *is an abstract rigidity matroid* $\Leftrightarrow$ *(P1)(P2)* $\Leftrightarrow$ *(P1)(P3)* $\Leftrightarrow$ *(P1)(P4)* $\Leftrightarrow$ *(P2)(P3)* $\Leftrightarrow$ *(P2)(P4).*

To state our hypergraph extension, we first need to extend several notations to hypergraphs. For positive integers n and r, let K_n^r be the *r-uniform complete hypergraph* on n vertices. The *r-uniform star hypergraph* S_n^r is an r-uniform hypergraph consisting of $n-(r-1)$ hyperedges sharing $r-1$ vertices in common. Note that this is a hypergraph extension of $K_{1,n-1}$.

There are several possible extensions of d-valent extension operation to hypergraphs, and in this work we adopt the following definition. For an r-uniform hypergraph G, a set X of $r-1$ vertices is said to be *empty* if G has no hyperedge that contains X. (We use this terminology even if some vertex in X is missing in G; in such a case, X is always empty in G.) Then a *d-valent extension* is an operation that creates a new hypergraph from G by adding d new distinct r-hyperedges (possibly by adding new vertices) such that those new hyperedges share $(r-1)$ vertices that form an empty $(r-1)$-set in G. See Figure 2.

We are now ready to state our main theorem.

Theorem 2 *Let n, r, d be positive integers with $r \geq 2, d \geq r-1$ and $n \geq d+2$, and let $\mathcal{M}$ be a matroid on the edge set of the r-uniform complete hypergraph with n vertiecs. Consider the following four properties:*

(P1) the rank of $\mathcal{M}$ is $\binom{n}{r} - \binom{n-(d-r+2)}{r}$;

(P2) the edge set of each subgraph isomorphic to K_{d+2}^r is a circuit;

(P3) the edge set of each subgraph isomorphic to S_n^r minus $d-r+1$ hyperedges is a cocircuit;

(P4) each $(d-r+2)$-valent extension preserves the independence of the edge sets of hypergraphs.

Then

$$\textit{(P1)(P2)} \Leftrightarrow \textit{(P1)(P3)} \Leftrightarrow \textit{(P1)(P4)} \Leftrightarrow \textit{(P2)(P3)} \Leftrightarrow \textit{(P2)(P4)}.$$

Motivated by Theorem 2 we define a matroid $\mathcal{M}$ on the edge set of K_n^r as an *abstract d-rigidity matroid* if $\mathcal{M}$ satisfies any equivalent pair of the properties listed in Theorem 2.

3 Skeletal Rigidity Matroids

A primary example of abstract rigidity matroids is the skeletal rigidity matroid. Let V be a finite set, d, r be positive integers with $d \geq r-1$. We use $\binom{V}{r}$ to denote the set of all subsets of size r in V. For a point-configuration $p : V \to \mathbb{R}^{d+1} \setminus \{0\}$, consider the matrix $R_{r,d}(V,p)$ of size $|\binom{V}{r}| \times \binom{d+1}{r}|\binom{V}{r-1}|$ such that each row is indexed by each element of $\binom{V}{r}$, each consecutive $\binom{d+1}{r}$ columns are indexed by each element of $\binom{V}{r-1}$, and the row of $\rho = \{v_1, \ldots, v_r\} \in \binom{V}{r}$ with $v_1 < \cdots < v_r$ is given by

$$\begin{array}{cccccccc} \cdots & & \rho \setminus \{v_1\} & \cdots & \rho \setminus \{v_i\} & \cdots & \rho \setminus \{v_r\} & \cdots \\ \big[\ 0 & p(v_1) \vee p(v_2) \vee \cdots \vee p(v_r) & 0 & p(v_i) \vee p(v_1) \vee \cdots \vee p(v_r) & 0 & p(v_r) \vee p(v_1) \vee \cdots \vee p(v_{r-1}) & 0\ \big], \end{array}$$

where $p_1 \vee \cdots \vee p_r$ denotes the exterior product of points $p_1, \ldots, p_r$, which is regarded as a $\binom{d+1}{r}$-dimensional vector by taking the standard basis. Since the rank of $R_{r,d}(V,p)$ is invariant over generic point-configurations p, one can define the *skeletal d-rigidity matroid* $\mathcal{R}_{n,r,d}$ on the edge set of K_n^r as the row matroid of $R_{r,d}([n],p)$ for a generic p.

Example 1. Consider the matrix $R_{3,3}([4],p)$. Each $p(v_1) \vee p(v_2) \vee p(v_3)$ for $\{v_1,v_2,v_3\} \in \binom{[4]}{3}$ is a 4-dimensional vector. We have the matrix $R_{3,3}([4],p)$ as follows, where $p_v = p(v)$ and blank entries are zero:

	$\{1,2\}$	$\{1,3\}$	$\{1,4\}$	$\{2,3\}$	$\{2,4\}$	$\{3,4\}$
$\{1,2,3\}$	$p_3 \vee p_1 \vee p_2$	$p_2 \vee p_1 \vee p_3$		$p_1 \vee p_2 \vee p_3$		
$\{1,2,4\}$	$p_4 \vee p_1 \vee p_2$		$p_2 \vee p_1 \vee p_4$		$p_1 \vee p_2 \vee p_4$	
$\{1,3,4\}$		$p_4 \vee p_1 \vee p_3$	$p_3 \vee p_1 \vee p_4$			$p_1 \vee p_3 \vee p_4$
$\{2,3,4\}$				$p_4 \vee p_2 \vee p_3$	$p_3 \vee p_2 \vee p_4$	$p_2 \vee p_3 \vee p_4$

Suppose p is generic. The rank of $R_{3,3}([4],p)$ is four, which means $\mathcal{R}_{4,3,3}$ is the free matroid on $\{1,2,3,4\}$.

Example 2. When $d = r-1$, each $p_1 \vee \cdots \vee p_r$ is a scalar, and $R_{r,r-1}(V,p)$ is row-equivalent to the boundary operator between $\binom{V}{r}$ and $\binom{V}{r-1}$ in the simplicial chain complex of $K^r_{|V|}$ (by regarding $K^r_{|V|}$ as a $(r-1)$-dimensional simplicial complex). Thus $\mathcal{R}_{n,r,r-1}$ corresponds to the graphic matroid of the complete r-uniform hypergraph.

Example 3. When $r = 2$, $R_{2,d}(V,p)$ corresponds to the projected version of the rigidity matrix of bar-joint frameworks due to Crapo and Whiteley [5], and it is equivalent to the ordinary rigidity matrix. Hence $\mathcal{R}_{n,2,d}$ coincides with the ordinary generic d-rigidity matroid.

It is a direct consequence of results of Tay, White, and Whiteley [15] that $\mathcal{R}_{n,r,d}$ satisfies (P1) in Theorem 2. We can check that (P4) holds in $\mathcal{R}_{n,r,d}$ by examining the row independence of $R_{r,d}([n],p)$.

Proposition 3 *Suppose n, r, d are as in Theorem 2. Then $\mathcal{R}_{n,r,d}$ is an abstract d-rigidity matroid.*

4 Cofactor Matroids

As the second example of abstract rigidity matroids, we introduce cofactor matroids of hypergraphs. Let V be a finite set, r be a positive integer, s be a non-negative integer, and $p : V \to \mathbb{R}^{r+1} \setminus \{0\}$ be a point-configuration. We use $p(v) = (p_{v,1}\ p_{v,2}\ \ldots\ p_{v,r+1})^\top \in \mathbb{R}^{r+1}$ to denote the coordinates of $p(v)$.

For each $\rho = \{v_1, \ldots, v_r\} \in \binom{V}{r}$ with $v_1 < \cdots < v_r$, a linear form $\ell_\rho(p)$ representing the linear subspace spanned by $\{p(v_1), \ldots, p(v_r)\}$ is given by

$$\ell_\rho(p) = \begin{vmatrix} p_{v_1,1} & p_{v_2,1} & \cdots & p_{v_r,1} & x_1 \\ p_{v_1,2} & p_{v_2,2} & \cdots & p_{v_r,2} & x_2 \\ \vdots & \vdots & & \vdots & \vdots \\ p_{v_1,r+1} & p_{v_2,r+1} & \cdots & p_{v_r,r+1} & x_{r+1} \end{vmatrix}$$

with indeterminates $x_1, x_2, \ldots, x_{r+1}$. The *cofactor matrix* $C_{r,s}(V,p)$ is defined as a matrix of size $|\binom{V}{r}| \times \binom{s+r}{s}|\binom{V}{r-1}|$ such that each row is indexed by each element of $\binom{V}{r}$, each consecutive $\binom{s+r}{s}$ columns are indexed by each element of $\binom{V}{r-1}$, and the row of $\rho \in \binom{V}{r}$ is given by

$$\begin{matrix} \cdots & \rho \setminus \{v_1\} & \cdots & \rho \setminus \{v_i\} & \cdots & \rho \setminus \{v_r\} & \cdots \\ [\ 0 & \mathrm{sign}(v_1,\rho)[(\ell_\rho(p))^s] & 0 & \mathrm{sign}(v_i,\rho)[(\ell_\rho(p))^s] & 0 & \mathrm{sign}(v_r,\rho)[(\ell_\rho(p))^s] & 0\ ], \end{matrix}$$

where $[(\ell_\rho(p))^s]$ is a $\binom{s+r}{s}$-dimensional tuple consisting of the coefficients of $\binom{s+r}{s}$ monomials of $(\ell_\rho)^s$ in $x_1, x_2, \ldots, x_{r+1}$ and $\mathrm{sign}(v_i,\rho)$ is either $+1$ or -1 according to the parity of the order of v_i in ρ. (Recall that the elements of ρ are ordered such that $v_1 < \cdots < v_r$.)

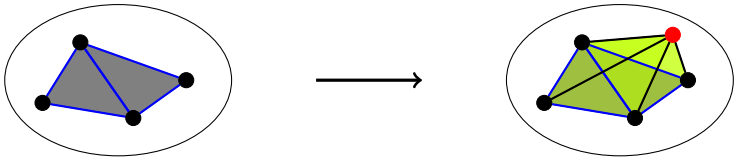


Figure 3: A coning operation on a 3-uniform hypergraph.

For example, when $s = 2$, $(\ell_\rho(p))^2$ is a homogeneous polynomial of degree two, which is written as $(\ell_\rho(p))^2 = \sum_{1\le i\le j\le r+1} c_{ij}x_ix_j$. Then $[(\ell_\rho(p))^2] = (c_{1,1}, c_{1,2}, \dots, c_{r+1,r+1})$.

Since the rank of $C_{r,s}(V,p)$ is invariant over generic point configurations p, one can define the *s-cofactor matroid* $\mathcal{C}_{n,r,s}$ on the edge set of K_n^r as the row matroid of $C_{r,s}([n],p)$ for a generic p.

Example 4. When $s = 0$, $[\ell_\rho(p)^0] = 1$ and hence $C_{r,0}(V,p)$ is the boundary operator between $\binom{V}{r}$ and $\binom{V}{r-1}$ in the simplicial chain complex of $K^r_{|V|}$. Thus, as in the case of skeletal $(r-1)$-rigidity, $\mathcal{C}_{n,r,0}$ is the graphic matroid of the complete r-uniform hypergraph.

Example 5. When $s = 1$, $[\ell_\rho(p)^1] = p(v_1) \vee \cdots \vee p(v_r)$ for every $\rho = \{v_1, \dots, v_r\}$, and hence $C_{r,1}(V,p) = R_{r,r}(V,p)$. So the 1-cofactor matroid $\mathcal{C}_{n,r,1}$ coincides with the skeletal r-rigidity matroid $\mathcal{R}_{n,r,r}$.

Unlike skeletal rigidity, there is no substantial research on this matroid since Whiteley introduced it in [16], and little are known about its basic properties. (This is mainly because in the context of splines one usually focuses on topologically defined graphs or simplicial complexes such as planar triangulations, and there are little motivation for looking at dense complexes.) Hence for cofactor matroids we need to check the conditions for abstract rigidity from scratch.

Our first observation for cofactor matroids is that independence is preserved by extension.

Lemma 4 *Let G, H be subgraphs of K_n^r such that H is obtained from G by $(s+1)$-valent extension. Then $E(G)$ is independent in $\mathcal{C}_{n,r,s}$ if and only if $E(H)$ is independent in $\mathcal{C}_{n,r,s}$.*

The proof is a direct adaptation of that of the corresponding statement for ordinary graphs in [16, Lemma 11.3.6].

By Theorem 2, to check the abstract rigidity, it remains to show that K^r_{s+r+1} is a circuit. This turns out to be nontrivial and we check it by showing the coning property first.

Given a r-uniform hypergraph G, its *cone* $G * v_0$ is defined as the new graph obtained from G by adding a new vertex v_0 and adding a new hyperedge $\tau \cup \{v_0\}$ for every $\tau \in \binom{V(G)}{r-1}$ with $\tau \subset \rho$ for some $\rho \in E(G)$. See Figure 3.

Theorem 5 *Let G and $G * v_0$ be subgraphs of K_n^r such that $G * v_0$ is the cone of G. Then $E(G)$ is independent in $\mathcal{C}_{n,r,s}$ if and only if $E(G * v_0)$ is independent in $\mathcal{C}_{n,r,s+1}$.*

The proof is a careful adaptation of that of the corresponding statement for ordinary graphs in [16, Theorem 11.3.3]. It should be also noted that Theorem 5 confirms Conjecture (e) in [16, Section 16.5].

Observe that K^r_{s+r+1} can be constructed from K^r_{r+2} by a sequence of coning operations. Since the cofactor matroid coincides with the skeletal rigidity matroid if $s = 1$ (by Example 5), the rank of K^r_{r+2} in $\mathcal{C}_{n,r,1}$ is equal to that in $\mathcal{R}_{n,r,r}$. Hence, the edge set of K^r_{s+r+1} is dependent by Theorem 5. Moreover, K^r_{s+r+1} minus one hyperedge can be constructed from an empty hypergraph by $(s+1)$-valent extensions (possibly allowing addition of less than $s+1$ new hyperedges in each extension). Thus we obtain the following by Lemma 4.

Lemma 6 *The edge set of K^r_{s+r+1} is a circuit in $\mathcal{C}_{n,r,s}$.*

Combining, Theorem 2, Lemma 4, and Lemma 6, we obtain our second main result.

Proposition 7 *Suppose n, r, s are integers with $r \ge 2$, $s \ge 0$, and $n \ge s + r + 1$. Then $\mathcal{C}_{n,r,s}$ is an abstract $(s+r-1)$-rigidity matroid.*

5 Maximality Conjecture

The central question in the context of abstract rigidity is the following conjecture of Graver.

Conjecture 8 (Graver [6]) *The generic 3-dimensional rigidity matroid (of ordinary graphs) is the unique maximal abstract 3-rigidity matroid.*

Since Clinch, Jackson, and Tanigawa [3] proved that the 2-cofactor matroid (of ordinary graphs) is the unique maximal abstract 3-rigidity matroid, the conjecture is equivalent to the following conjecture of Whiteley.

Conjecture 9 (Whiteley [16]) *The generic 3-dimensional rigidity matroid and the 2-cofactor matroid coincide for ordinary graphs.*

The corresponding question for other dimension d has been already answered. When $d \leq 2$, the generic d-dimensional rigidity matroid coincides with the $(d-1)$-cofactor matroid (due to the classical Maxwell-Cremona correspondence) and it is the unique maximal abstract d-rigidity matroid [6]. On the other hand, when $d \geq 4$, the generic d-dimensional rigidity matroid is distinct from the $(d-1)$-cofactor matroid [16]. In particular, the rank of the edge set of $K_{d+2,d+2}$ becomes strictly smaller in the generic d-dimensional rigidity matroid.

We now extend this investigation to hypergraphs. For hypergraphs, a natural target would be a complete partite hypergraph. Let $K^r_{i_1,i_2,\dots,i_r}$ be the complete r-partite r-uniform hypergraph consisting of disjoint vertex sets $V_1, \dots, V_r$ with $|V_j| = i_j$.

Lemma 10 *For any n, d, r with $n \geq (d+2)r$, $d \geq r+1$, and $r \geq 2$, $K^r_{d+2,d+2,\dots,d+2}$ is dependent in $\mathcal{R}_{n,r,d}$*

In order to show the difference between the skeletal rigidity matroid and the cofactor matroid, our next goal is to show the independence of $K^r_{d+2,d+2,\dots,d+2}$ in $\mathcal{C}_{n,r,d-r+1}$. Currently we have not yet succeeded in proving this. However, for fixed (small) d, this can be checked by directly computing the rank of the cofactor matrix by picking a random point configuration p. (Note that this computational experiment indeed gives a mathematical proof since rank $C_{r,s}(V,p) \leq$ rank $C_{r,s}(V,q)$ for any $p : V \to \mathbb{R}^{d+1} \setminus \{0\}$ and any generic $q : V \to \mathbb{R}^{d+1}$.) The result of the computational experiment shows that the edge set of $K^r_{d+2,d+2,\dots,d+2}$ is independent in $\mathcal{C}_{n,r,d-r+1}$ for $r \in \{3,4,5\}$ and $d = r+1$. Hence, by Lemma 10, we have $\mathcal{R}_{n,r,d} \neq \mathcal{C}_{n,r,d-r+1}$ for those r and d. We can use coning operations to construct further examples that separate $\mathcal{R}_{n,r,d}$ and $\mathcal{C}_{n,r,d-r+1}$ for all $d \geq r+1$. Summarizing observations so far, we have the following.

Theorem 11 *Let n, d, r be positive integers with $n \geq r(d+2)$ and $d \geq r-1$. Suppose $r \in \{3,4,5\}$. Then $\mathcal{R}_{n,r,d} = \mathcal{C}_{n,r,d-r+1}$ if and only if $d \in \{r-1, r\}$.*

Theorem 11 verifies and modifies[1] conjectures of Whiteley [16, Figure 13.1]. In view of our computation experiment, it is likely that Theorem 11 is true for any $r \geq 3$ but we do not know how to prove it. See Figure 4.

Currently we do not have any example that refutes the following maximality conjecture by Whiteley.

Conjecture 12 (Whiteley [16]) *The s-cofactor matroid $\mathcal{C}_{n,r,s}$ is the unique maximal abstract $(s+r-1)$-rigidity matroid on the edge set of K^r_n.*

It should be noted that, even if $s \leq 1$ and $r \geq 3$, the conjecture is open.

References

[1] K. Adiprasito, Combinatorial Lefschetz theorems beyond positivity, arXiv:1812.10454 (2018)

[1] Whiteley [16, Figure 13.1] conjectured that $\mathcal{R}_{n,r,d} = \mathcal{C}_{n,r,d-1}$ if $d = r+1$, but this is not the case if $r \geq 3$.

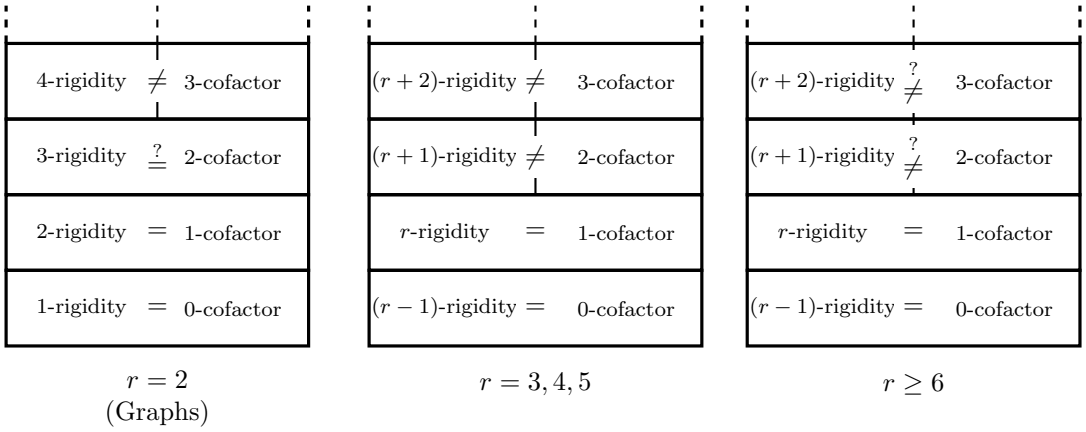


Figure 4: The schematic diagram between the skeletal d-rigidity matroids and the s-cofactor matroids of the complete r-uniform hypergraph.

[2] L. J. Billera, Homology of smooth splines: generic triangulations and a conjecture of Strang, *Transactions of the American Mathematical Society* **310** (1988), 325–340.

[3] K. Clinch, B. Jackson, and S. Tanigawa, Abstract 3-rigidity and bivariate C_2^1-splines I: Whiteley's maximality conjecture, *Discrete Analysis* **2022:2** (2022), 50 pp.

[4] K. Clinch, B. Jackson, and S. Tanigawa, Abstract 3-rigidity and bivariate C_2^1-splines II: Combinatorial characterization, *Discrete Analysis* **2022:3** (2022), 32 pp.

[5] H. Crapo and W. Whiteley, Statics of frameworks and motions of panel structures, a projective geometric introduction, *Structural Topology* **6** (1982), 43–82.

[6] J. E. Graver, Rigidity matroids, *SIAM Journal on Discrete Mathematics* **4** (1991), 355–368.

[7] J. E. Graver, B. Servatius, and H. Servatius, Abstract rigidity in m-space, in: *Jerusalem Combinatorics '93*, American Mathematical Society (1994), 145–151.

[8] C. W. Lee, Generalized stress and motions, in: *Polytopes: abstract, convex and computational*, Springer, Dordrecht (1994), 249–271.

[9] C. W. Lee, P.L.-spheres, convex polytopes, and stress, *Discrete & Computational Geometry* **15** (1996), 389–421.

[10] V-H. Nguyen, On abstract rigidity matroids, *SIAM Journal on Discrete Mathematics* **24** (2010), 363–369.

[11] I. Novik and H. Zheng, Reconstructing simplicial polytopes from their graphs and affine 2-stresses, arXiv:2106.09284 (2021)

[12] R. P. Stanley, The number of faces of a simplicial convex polytope, *Advances in Mathematics* **35** (1980), 236–238.

[13] T-S. Tay, N. White, and W. Whiteley, Skeletal rigidity of simplicial complexes, I, *European Journal of Combinatorics* **16** (1995), 381–403.

[14] T-S. Tay, N. White, and W. Whiteley, Skeletal rigidity of simplicial complexes, II, *European Journal of Combinatorics* **16** (1995), 503–523.

[15] T-S. Tay and W. Whiteley, A homological interpretation of skeletal rigidity, *Advances in Applied Mathematics* **25** (2000), 102–151.

[16] W. Whiteley, Some matroids from discrete applied geometry, in: *Matroid Theory*, American Mathematical Society (1996), 171–312.

A combinatorial algorithm for computing the entire sequence of the maximum degree of minors of a generic partitioned polynomial matrix with 2×2 submatrices[1]

Yuni Iwamasa

Graduate School of Informatics
Kyoto University
Kyoto 606-8501, Japan
iwamasa@i.kyoto-u.ac.jp

Abstract: In this paper, we consider the problem of computing the entire sequence of the maximum degree of minors of a block-structured symbolic matrix $A = (A_{\alpha\beta} x_{\alpha\beta} t^{d_{\alpha\beta}})$, where $A_{\alpha\beta}$ is a 2×2 matrix over a field $\mathbf{F}$, $x_{\alpha\beta}$ is an indeterminate, and $d_{\alpha\beta}$ is an integer for $\alpha = 1, 2, \ldots, \mu$ and $\beta = 1, 2, \ldots, \nu$, and t is an additional indeterminate. The main result is a combinatorial $O(\mu\nu \min\{\mu, \nu\}^2)$-time algorithm for the above problem. We also present a minimax theorem, which can be used as a good characterization (NP $\cap$ co-NP characterization) for the problem.

Keywords: Generic partitioned polynomial matrix, Degree of minor, Weighted Edmonds' problem, Weighted non-commutative Edmonds' problem

1 Introduction

It is well-known that the Hungarian method [7], which is a classical primal-dual algorithm for the maximum weight bipartite matching problem, can output a bipartite matching of size k having maximum weight among all matchings with the same size for all possible values of k. This fact has the following algebraic interpretation: For a bipartite graph $G = (\{1, 2, \ldots, m\}, \{1, 2, \ldots, n\}; E)$ with edge weights d_{ij} for $ij \in E$, the Hungarian method computes the entire sequence of the maximum degree of minors of $A(t)$ defined by $(A(t))_{ij} \coloneqq x_{ij} t^{d_{ij}}$ if $ij \in E$ and zero otherwise, where x_{ij} is a variable for each edge ij and t is another variable. Indeed, the maximum weight of a matching of size k in G is equal to the maximum degree $\delta_k(A(t))$ of the minors of order k, i.e.,

$$\delta_k(A(t)) \coloneqq \sup\{\deg \det B(t) \mid B(t)\colon\ k \times k \text{ submatrix of } A(t)\},$$

where the determinant $\det B(t)$ of $B(t)$ is regarded as a polynomial in t and $\delta_0(A(t)) \coloneqq 0$. Thus, the entire sequence $\big(\delta_0(A(t)), \delta_1(A(t)), \ldots, \delta_{\min\{m,n\}}(A(t))\big)$ of the maximum degree of minors equals the sequence of the maximum weights of a matching of size k for $k = 0, 1, \ldots, \min\{m, n\}$; the Hungarian method computes this.

In this paper, we consider the (2×2)-analog of the above problem, i.e., the problem of computing the entire sequence of the maximum degree of minors of the following (2×2)-block-structured matrix:

$$A(t) = \begin{pmatrix} A_{11} x_{11} t^{d_{11}} & A_{12} x_{12} t^{d_{12}} & \cdots & A_{1\nu} x_{1\nu} t^{d_{1\nu}} \\ A_{21} x_{21} t^{d_{21}} & A_{22} x_{22} t^{d_{22}} & \cdots & A_{2\nu} x_{2\nu} t^{d_{2\nu}} \\ \vdots & \vdots & \ddots & \vdots \\ A_{\mu 1} x_{\mu 1} t^{d_{\mu 1}} & A_{\mu 2} x_{\mu 2} t^{d_{\mu 2}} & \cdots & A_{\mu\nu} x_{\mu\nu} t^{d_{\mu\nu}} \end{pmatrix}. \tag{1}$$

[1] A preliminary version of this extended abstract is [5] and its full version is available at [6]. This research is supported by JSPS KAKENHI Grant Numbers JP17K00029, 20K23323, 20H05795, 22K17854, Japan.

Here $A_{\alpha\beta}$ is a 2×2 matrix over a field $\mathbf{F}$, $x_{\alpha\beta}$ is a variable, and $d_{\alpha\beta}$ is an integer for $\alpha = 1, 2, \ldots, \mu$ and $\beta = 1, 2, \ldots, \nu$, and t is another variable. Note that that, if we replace each 2×2 matrix $A_{\alpha\beta}$ with a 1×1 matrix (or a scalor), then the resulting (esstentially) coincides with the matrix obtained from a bipartite graph described above. A matrix $A(t)$ of the form (1) is called a (2×2)-*type generic partitioned polynomial matrix*.

This problem has been studied in the context of *weighted Edmonds' problem* and *weighted noncommutative Edmonds' problem* [3]; weighted Edmonds' problem asks to compute the entire sequence of the maximum degree of minors of

$$A(t) = A_1(t)x_1 + A_2(t)x_2 + \cdots + A_\ell(t)x_\ell,$$

where $A_k(t)$ is a polynomial matrix over a field $\mathbf{F}$ with an indeterminate t; in weighted noncommutative Edmonds' problem, we suppose that the variables x_i and x_j are noncommutative but t is commutative for any variable x_i. It is known [9, 8, 1] that weighted noncommutative Edmonds' problem is an algebraic generalization of weighted bipartite matching and weighted linear matroid intersection, and that weighted Edmonds' problem additionally generalizes weighted nonbipartite matching and weighted linear matroid parity. Although the polynomial-time solvability of weighted (noncommutative) Edmonds' problem is not known, Hirai and Ikeda [4] introduce a strongly polynomial-time solvable subclass of weighted noncommutative Edmonds' problem, which contains our problem. That is, the strongly polynomial-time solvability of our problem has already been known. Their polynomial-time algorithm is conceptually simple, but is slow and not combinatorial.

The main result of the present article is a faster and combinatorial polynomial-time algorithm for our problem, which can be viewed as an algebraic generalization of the Hungarian method:

Theorem 1 *Let $A(t)$ be a (2×2)-type generic partitioned polynomial matrix of the form (1). There exists a combinatorial $O(\mu\nu \min\{\mu, \nu\}^2)$-time algorithm for computing the entire sequence of the maximum degree of minors of $A(t)$.*

Our algorithm is based on a new duality theorem on the degree of the determinant of a (2×2)-type generic partitioned polynomial matrix $A(t)$ established in this study; this duality theorem can be viewed as an algebraic generalization of Egerváry's theorem [2] that is a minimax theorem for the weighted bipartite matching problem.

The algorithm description and all proofs are omitted; see [6] for the full version.

Notations. For a positive integer k, we denote $\{1, 2, \ldots, k\}$ by $[k]$. Let $A(t)$ be a (2×2)-type generic partitioned polynomial matrix of the form (1). The matrix $A(t)$ is regarded as a matrix over the field $\mathbf{F}(x, t)$ of rational functions with variables t and $x_{\alpha\beta}$ for $\alpha \in [\mu]$ and $\beta \in [\nu]$. The symbols α, β, and γ are used to represent a row-block index in $[\mu]$, column-block index in $[\nu]$, and row- or column-block index in $[\mu] \sqcup [\nu]$ of $A(t)$, respectively, where $\sqcup$ denotes the direct sum. We often drop "$\in [\mu]$" from the notation of "$\alpha \in [\mu]$" if it is clear from the context. Each α and β is endowed with the 2-dimensional $\mathbf{F}$-vector space $\mathbf{F}^2$, denoted by U_α and V_β, respectively. Each submatrix $A_{\alpha\beta}$ is considered as the bilinear map $U_\alpha \times V_\beta \to \mathbf{F}$ defined by $A_{\alpha\beta}(u, v) \coloneqq u^\top A_{\alpha\beta} v$ for $u \in U_\alpha$ and $v \in V_\beta$. We denote by $\ker_{\mathrm{L}}(A_{\alpha\beta})$ and $\ker_{\mathrm{R}}(A_{\alpha\beta})$ the left and right kernels of $A_{\alpha\beta}$, respectively. Let us denote by $\mathcal{M}_\alpha$ and $\mathcal{M}_\beta$ the sets of 1-dimensional vector subspaces of U_α and V_β, respectively.

2 Minimax theorem

This section is devoted to presenting the minimax theorem for our problem. We first introduce a matching concept named *pseudo-matching* and a potential concept, which correspond to a primal and dual concept, respectively. An edge subset $M \subseteq E$ is called a *pseudo-matching* if it satisfies the following combinatorial and algebraic conditions (Deg), (Cycle), and (VL):

(Deg) $\deg_M(\gamma) \leq 2$ for each node γ of G.

Suppose that M satisfies (Deg). Then each connected component of M forms a path or a cycle. Thus M is 2-edge-colorable; i.e., there are two edge classes such that any two incident edges are in different classes. An edge in one color class is called a *+-edge*, and an edge in the other color class is called a *−-edge*.

(Cycle) Each cycle component of M has at least one rank-1 edge.

A *valid labeling* $\mathcal{V} = (\{U_\alpha^+, U_\alpha^-\}, \{V_\beta^+, V_\beta^-\})_{\alpha,\beta}$ is a node-labeling that assigns two distinct 1-dimensional subspaces to each node, $U_\alpha^+, U_\alpha^- \in \mathcal{M}_\alpha$ with $U_\alpha^+ \neq U_\alpha^-$ for α and $V_\beta^+, V_\beta^- \in \mathcal{M}_\beta$ with $V_\beta^+ \neq V_\beta^-$ for β, such that it satisfies, for each edge $\alpha\beta \in M$,

$$A_{\alpha\beta}(U_\alpha^+, V_\beta^-) = A_{\alpha\beta}(U_\alpha^-, V_\beta^+) = \{0\},$$
$$(\ker_{\mathrm{L}}(A_{\alpha\beta}), \ker_{\mathrm{R}}(A_{\alpha\beta})) = \begin{cases} (U_\alpha^+, V_\beta^+) & \text{if } \alpha\beta \text{ is a rank-1 } +\text{-edge}, \\ (U_\alpha^-, V_\beta^-) & \text{if } \alpha\beta \text{ is a rank-1 } -\text{-edge}. \end{cases}$$

(VL) M admits a valid labeling.

The *size* of a matching-pair (M, I) is $|M| + |I|$, and its *weight* $w(M, I)$ is

$$w(M, I) \coloneqq \sum_{\alpha\beta \in M} d_{\alpha\beta} + \sum_{\alpha\beta \in I} d_{\alpha\beta}.$$

For $c \in \mathbf{R}$, a function $p : \bigcup_\gamma \mathcal{M}_\gamma \to \mathbf{R}$ is called a *c-potential* if

- p is nonnegative, i.e., $p(Z) \geq 0$ for all $Z \in \bigcup_\gamma \mathcal{M}_\gamma$, and
- $p(X) + p(Y) + c \geq d_{\alpha\beta}$ for all $\alpha\beta \in E$, $X \in \mathcal{M}_\alpha$, and $Y \in \mathcal{M}_\beta$ such that $A_{\alpha\beta}(X, Y) \neq \{0\}$.

The following is our minimax formula:

Theorem 2 *Let k be a nonnegative integer. The following values* (i)–(iii) *are the same:*

(i) $\delta_k(A(t))$.

(ii) $\sup\{w(M, I) \mid (M, I)$*: matching-pair of size* $k\}$.

(iii) $\inf\{p(\mathcal{V}) + kc \mid \mathcal{V}$*: labeling,* $c \in \mathbf{R}$*,* p*: c-potential*$\}$.

For using the above duality theorem as a good characterization for our problem, we introduce the concept of compatibility as follows. Let (M, I) be a matching-pair of size k and $\mathcal{V}$ a valid labeling for M. A c-potential p is said to be $(M, I, \mathcal{V})$-*compatible* if p satisfies the following conditions (Reg) and (Tight):

(Reg) For each α and β,

$$p(X) = \max\{p(U_\alpha^+), p(U_\alpha^-)\} \qquad (X \in \mathcal{M}_\alpha \setminus \{U_\alpha^+, U_\alpha^-\}),$$
$$p(Y) = \max\{p(V_\beta^+), p(V_\beta^-)\} \qquad (Y \in \mathcal{M}_\beta \setminus \{V_\beta^+, V_\beta^-\}).$$

(Tight) For each $\alpha\beta \in M$,

$$d_{\alpha\beta} = \begin{cases} p(U_\alpha^-) + p(V_\beta^-) + c & \text{if } \alpha\beta \text{ is a } +\text{-edge}, \\ p(U_\alpha^+) + p(V_\beta^+) + c & \text{if } \alpha\beta \text{ is a } -\text{-edge}, \end{cases}$$

and for each $\alpha\beta \in I$,

$$d_{\alpha\beta} = \begin{cases} p(U_\alpha^+) + p(V_\beta^+) + c & \text{if } \alpha\beta \text{ is a } +\text{-edge}, \\ p(U_\alpha^-) + p(V_\beta^-) + c & \text{if } \alpha\beta \text{ is a } -\text{-edge}. \end{cases}$$

An $(M, I, \mathcal{V})$-compatible c-potential p is said to be *optimal* if the equality $w(M, I) = p(\mathcal{V}) + kc$ holds, namely, (M, I) and $(p, \mathcal{V})$ attain the supremum of (ii) and the infimum of (iii) in Theorem 2, respectively.

Theorem 3 *Let k be a nonnegative integer. If $\delta_k(A(t))$ is bounded, then there are a matching-pair (M, I) of size k, a valid labeling $\mathcal{V}$ for M, and an optimal $(M, I, \mathcal{V})$-compatible c-potential p for some $c \in \mathbf{R}$. In particular, the above p and c can be chosen to be integer-valued.*

By Theorem 3, a pair $(p, \mathcal{V})$ of a c-potential p satisfying (Reg) and a valid labeling $\mathcal{V}$ satisfying $p(\mathcal{V}) + kc < \theta$ can be used as a proof for $\delta_k(A(t)) < \theta$. Furthermore, the condition (Reg) enables us to check if a given nonnegative function p on $\bigcup_\gamma \mathcal{M}_\gamma$ is a c-potential in polynomial time. Thus the proof $(p, \mathcal{V})$ for $\delta_k(A(t)) < \theta$ is verifiable in polynomial time, implying the problem of whether $\delta_k(A(t)) \geq \theta$ is in co-NP.

References

[1] P. M. Camerini, G. Galbiati, and F. Maffioli. Random pseudo-polynomial algorithms for exact matroid problems. *Journal of Algorithms*, 13:258–273, 1992.

[2] J. Egerváry. Matrixok kombinatorius tulajdonságairól. *Matematikai és Fizikai Lapok*, pages 16–28, 1931.

[3] H. Hirai. Computing the degree of determinants via discrete convex optimization on Euclidean buildings. *SIAM Journal on Applied Algebra and Geometry*, 3(3):523–557, 2019.

[4] H. Hirai and M. Ikeda. A cost-scaling algorithm for computing the degree of determinants. *Computational Complexity*, 31:Article 10, 2022.

[5] Y. Iwamasa. A combinatorial algorithm for computing the degree of the determinant of a generic partitioned polynomial matrix with 2×2 submatrices. In *Proceedings of the 22nd Conference on Integer Programming and Combinatorial Optimization (IPCO 2021)*, volume 12707 of *LNCS*, pages 119–133, 2021.

[6] Y. Iwamasa. A combinatorial algorithm for computing the entire sequence of the maximum degree of minors of a generic partitioned polynomial matrix with 2×2 submatrices. arXiv:2104.14841v2, 2021.

[7] H. W. Kuhn. The Hungarian method for assignment problems. *Naval Research Logistics Quarterly*, 2:83–97, 1955.

[8] L. Lovász. Singular spaces of matrices and their application in combinatorics. *Boletim da Sociedade Brasileira de Matemática*, 20(1):87–99, 1989.

[9] W. T. Tutte. The factorization of linear graphs. *Journal of the London Mathematical Society*, 22(2):107–111, 1947.

Openly Disjoint Paths, Jump Systems, and Discrete Convexity

Satoru Iwata[1]

Department of Mathematical Informatics
University of Tokyo
Tokyo 113-8656, Japan
iwata@mist.i.u-tokyo.ac.jp

Yu Yokoi[2]

National Institute of Informatics
Tokyo 101-8430, Japan
yokoi@nii.ac.jp

Abstract: Let G be a graph with a specified set T of terminals. In 1978, Mader discovered a min-max theorem on the number of openly disjoint T-paths. Sadli and Sebő (2000) showed that the set of integer vectors in $\mathbb{Z}^T$ that appear as degree sequences of openly disjoint T-paths forms a jump system. In this paper, we describe an alternative proof of this fact by using a reduction to matroid matching, which was originally observed by Lovász (1980). In addition, we show that a function on this jump system determined by the minimum total length of openly disjoint T-paths enjoys the M-convexity introduced by Murota (2006).

Keywords: Openly disjoint paths, delta-matroid, jump system, M-convex function

1 Introduction

Let $G = (V, E)$ be a graph with a specified set $T \subseteq V$ of terminals. A path in G is called a T-path if its ends are distinct vertices in T and no internal vertices belong to T. Two T-paths are called openly disjoint if they do not share any internal vertices. In 1978, Mader [16] showed a characterization of the maximum number of openly disjoint T-paths. The result contains as its special case Gallai's min-max theorem on the maximum number of vertex-disjoint T-paths [10], which is equivalent to the Tutte-Berge formula on maximum matching [1, 30], and Mader's min-max theorem [15] on edge-disjoint T-paths, which extends the theorem of Lovász [12] and Cherkassky [5] on edge-disjoint T-paths in inner Eulerian graphs.

Lovász [13] then introduced an equivalent variant, called disjoint $\mathcal{S}$-paths, where $\mathcal{S}$ is a given partition of the terminals, to provide an alternative proof via the matroid matching theorem. See also [29] for a minor correction. Schrijver [24] provided a short alternative proof for Mader's theorem on disjoint $\mathcal{S}$-paths based on Gallai's min-max theorem. Analogously to the Edmonds–Gallai decomposition for maximum matching, Sebő and Szegő [28] introduced a canonical decomposition that captures all the maximum disjoint $\mathcal{S}$-paths.

Schrijver [25] described a reduction of the disjoint $\mathcal{S}$-paths problem to the linear matroid parity problem. Consequently, one can use efficient linear matroid parity algorithms [6, 9, 20, 21] for finding the maximum number of disjoint $\mathcal{S}$-paths (or openly disjoint T-paths). The current best running time bound is $O(n^\omega)$, where n is the number of vertices and ω is the exponent of the fast matrix multiplications. This bound is achieved by the randomized algebraic algorithm of Cheung, Lau, and Leung [6]. The best deterministic running time bound due to Gabow and Stallmann [9] is $O(mn^\omega)$, where m is the number of edges. Without using the reduction to linear matroid parity, Chudnovsky, Cunningham, and Geelen [7] devised a combinatorial algorithm that runs in $O(n^5)$ time.

[1]Also affiliated at ICReDD, Hokkaido University, Sapporo, 001-0021, Japan. Research is supported by Grant-in-Aid for Scientific Research 20H05965 from JSPS and ERATO JPMJER1903 from JST.

[2]Research is supported by JST PRESTO Grant Number JPMJPR212B.

A natural weighted version of this setting is to find shortest disjoint $\mathcal{S}$-paths of a specified number. Yamaguchi [32] presented a reduction of this problem to the weighted linear matroid parity problem, which can be solved in polynomial time [11, 23].

This paper aims at clarifying discrete structures behind efficient solvability of these problems. Already in 2000, Sadli and Sebő [26] showed that the set of degree sequences of openly disjoint T-paths forms a jump system. See [27] in this volume for its proof. The concept of jump systems was introduced by Bouchet and Cunningham [3], as a generalization of delta-matroids of Bouchet [2], which are equivalent to pseudomatroids of Chandrasekaran and Kabadi [4]. In this paper, we provide an alternative proof of the result of Sadli and Sebő based on the reduction to linear matroid parity.

Extending the notion of M-convex functions in discrete convex analysis [18], Murota [19] introduced M-convex functions on jump systems. For an integer vector x, let $f(x)$ be the minimum total length of the openly disjoint T-paths whose degree sequence coincides with x. We show that this function f is an M-convex function on the jump system.

2 Jump Systems

For a pair of integer vectors x and y, we denote by $[x, y]$ the unique minimal box that contains both x and y, i.e., $[x, y] := \{\, z \mid z \in \mathbb{Z}^T, \forall t \in T, \min\{x(t), y(t)\} \le z(t) \le \max\{x(t), y(t)\} \,\}$. The distance between x and y is defined by $\mathrm{dist}(x, y) := \sum_{t\in T} |x(t) - y(t)|$. A *step* from x towards y is an integer vector $z \in [x, y]$ such that $\mathrm{dist}(x, z) = 1$. An integer vector $a \in \mathbb{Z}^T$ is said to be an (x, y)-*increment* if $x + a$ is a step from x towards y.

A *jump system*, introduced by Bouchet and Cunningham [3], is a set J of integer vectors that satisfies the following axiom.

(JS) For any $x, y \in J$ and an arbitrary step z from x towards y, either z belongs to J or J contains a step from z towards y.

For a graph $G = (V, E)$ with a terminal set $T \subseteq V$, an integer vector $x \in {\mathbb{Z}_+}^T$ is said to be *feasible* if there exists a family $\mathcal{P}$ of openly disjoint T-paths such that the number of T-paths in $\mathcal{P}$ incident to $t \in T$ coincides with $x(t)$ for every $t \in T$. Sadli and Sebő showed that the set of feasible integer vectors forms a jump system.

Proposition 2.1 (Sadli and Sebő [26, 27]) *The set J_G of feasible integer vectors for a graph $G = (V, E)$ with a terminal set $T \subseteq V$ forms a jump system.*

Since each T-path contributes two to the sum of the degrees at the terminals, $\sum_{t\in T} x(t)$ must be even for any feasible integer vector $x \in J_G$. Such a constant parity jump system is known to satisfy a stronger exchange property.

(JS*) For any $x, y \in J$ and an arbitrary (x, y)-increment a, there exists an $(x + a, y)$-increment b such that both $x + a + b$ and $y - a - b$ belong to J.

A function $f : J \to \mathbb{R}$ on a constant parity jump system is called an M-convex function if it satisfies the following property.

(M-JS) For any $x, y \in J$ and an arbitrary (x, y)-increment a, there exists an $(x + a, y)$-increment b such that $f(x) + f(y) \ge f(x + a + b) + f(y - a - b)$ holds.

Murota [19] showed that minimization problems on M-convex functions can be solved efficiently by greedy algorithms, provided that a function evaluation oracle is available.

Let $\ell : E \to \mathbb{R}_+$ be a nonnegative function that returns the length of each edge in G. Given a family $\mathcal{P}$ of openly disjoint T-paths in G, we denote by $\lambda(\mathcal{P})$ the total length of the T-paths in $\mathcal{P}$, i.e., $\lambda(\mathcal{P}) := \sum_{P\in\mathcal{P}} \sum_{e\in E(P)} \ell(e)$. For a feasible vector $x \in J_G$, let $f_G(x)$ denote the minimum value of $\lambda(\mathcal{P})$ for a family $\mathcal{P}$ of openly disjoint T-paths such that the number of paths in $\mathcal{P}$ incident to $t \in T$ coincides with $x(t)$ for every $t \in T$. Then f_G is a function on the set J_G of feasible vectors. The main contribution of this paper is to show that f_G is an M-convex function on J_G.

3 Delta-matroids

Delta-matroids are exactly the jump systems consisting of 0-1 vectors. A delta-matroid is a pair $(S, \mathcal{F})$ of a finite set S and a family $\mathcal{F}$ of subsets of S satisfying the following axiom.

(DM) For any $X, Y \in \mathcal{F}$ and an arbitrary $s \in X \triangle Y$, there exists an element $t \in X \triangle Y$ such that $X \triangle \{s, t\} \in \mathcal{F}$.

In particular, an even delta-matroid is a delta-matroid $(S, \mathcal{F})$ such that $|X \triangle Y|$ are even for all $X, Y \in \mathcal{F}$. Even delta-matroids can be characterized by the following exchange axiom.

(EDM) For any $X, Y \in \mathcal{F}$ and an arbitrary $s \in X \triangle Y$, there exists an element $t \in X \triangle Y \setminus \{s\}$ such that $X \triangle \{s, t\} \in \mathcal{F}$.

It is known that even delta-matroids enjoy the following simultaneous exchange property.

(SDM) For any $X, Y \in \mathcal{F}$ and an arbitrary $s \in X \triangle Y$, there exists an element $t \in X \triangle Y \setminus \{s\}$ such that $X \triangle \{s, t\} \in \mathcal{F}$ and $Y \triangle \{s, t\} \in \mathcal{F}$.

For a delta-matroid $(S, \mathcal{F})$ and a subset $Z \subseteq S$, consider the set family $\mathcal{F}_Z$ defined by $\mathcal{F}_Z := \{ X \mid X \subseteq S \setminus Z, X \cup Z \in \mathcal{F} \}$. Then $(S \setminus Z, \mathcal{F}_Z)$ forms a delta-matroid, which is called a contraction of $(S, \mathcal{F})$ by Z. In particular, if $(S, \mathcal{F})$ is an even delta-matroid, its contraction $(S, \mathcal{F}_Z)$ is also even.

A function $\omega : \mathcal{F} \to \mathbb{R}$ is called an valuation of an even delta-matroid $(S, \mathcal{F})$ if it satisfies the following axiom [8, 31].

(VDM) For any $X, Y \in \mathcal{F}$ and an arbitrary $s \in X \triangle Y$, there exists an element $t \in X \triangle Y \setminus \{s\}$ such that $\omega(X) + \omega(Y) \leq \omega(X \triangle \{s, t\}) + \omega(Y \triangle \{s, t\})$.

A pair of an even delta-matroid and its associated valuation is called a valuated delta-matroid.

A primary example of an even delta-matroid comes from alternating matrices. Let K be an alternating matrix whose row/column set is indexed by S. For a subset $X \subseteq S$, we denote by $K[X]$ the principal submatrix determined by X.

Lemma 3.1 *For an alternating matrix K whose row/column set is indexed by S, let $\mathcal{F}(K)$ be the family of subsets X of S such that $K[X]$ are nonsingular. Then $(S, \mathcal{F}(K))$ forms an even delta-matroid.*

The Pfaffian of an alternating matrix K is defined by

$$\operatorname{Pf} K := \sum_{\pi} \sigma_\pi \prod_{\{u,v\} \in \pi} K_{uv},$$

where the sum is taken over all partitions π of the row/column set into pairs and σ_π takes ± 1 in a suitable manner, see [14]. It is well-known that $\det K = (\operatorname{Pf} K)^2$ holds.

A primary example of a valuated matroid comes from alternating polynomial matrix.

Lemma 3.2 *For an alternating polynomial matrix $K(\theta)$ whose row/column set is indexed by S, let $\mathcal{F}(K)$ be the family of subsets that determine nonsingular principal submatrices of $K(\theta)$. Then $\omega : \mathcal{F}(K) \to \mathbb{Z}$ defined by $\omega(X) := \deg \operatorname{Pf} K(\theta)[X]$ is a valuation of the even delta-matroid $(S, \mathcal{F}(K))$.*

4 Disjoint $\mathcal{S}$-paths

To provide an alternative proof of Mader's theorem, Lovász [13] introduced an equivalent variant. Suppose that a terminal set $S \subseteq V$ is partitioned into a family $\mathcal{S}$ of disjoint subsets $S_1, \ldots, S_h$, i.e., $S = S_1 \cup \cdots \cup S_h$ and $S_i \cap S_j = \emptyset$ $(i \neq j)$. A path in G is called an $\mathcal{S}$-path if its ends belong to distinct members of $\mathcal{S}$ and no internal vertices belong to S. Mader's theorem can be reformulated to characterize the maximum number of disjoint $\mathcal{S}$-paths.

The matroid matching problem is a common generalization of matching and matroid intersection. Let $\rho : 2^E \to \mathbb{Z}$ be a monotone submodular function that satisfies $0 \leq \rho(F) \leq 2|F|$ for all $F \subseteq E$. A subset $F \subseteq E$ is called a matching if $\rho(F) = 2|F|$ holds. The matroid matching problem asks for finding a matching of maximum cardinality. Lovász [13] observed that finding the maximum number of disjoint $\mathcal{S}$-paths can be reduced to the matroid matching problem.

Following Schrijver [25, p. 1284], we now describe this reduction. Let E^+ and E^- be disjoint copies of E. The copies of $e \in E$ are denoted by $e^+ \in E^+$ and $e^- \in E^-$. Similarly, each vertex $v \in V \setminus S$ has two distinct copies v^+ and v^-. The set of these copies are defined by $U^+ := \{ v^+ \mid v \in V \setminus S \}$ and $U^- := \{ v^- \mid v \in V \setminus S \}$.

Consider a matrix A whose rows and columns are indexed respectively by $S \cup U^\pm$ and $E^\pm$, where $U^\pm := U^+ \cup U^-$ and $E^\pm := E^+ \cup E^-$. For any subsets $X \subseteq S \cup U^\pm$ and $Y \subseteq E^\pm$, we denote by $A[X, Y]$ the submatrix of A with row set X and column set Y. For each edge $e \in E$ we assign an arbitrary orientation. The tail and head of e are denoted by $\partial^+ e$ and $\partial^- e$, respectively. For each $t \in S_j$ and $e \in E$ with $t = \partial^+ e$, we put $A_{te^+} := 1$ and $A_{te^-} := j$. For each $t \in S_j$ and $e \in E$ with $t = \partial^- e$, we put $A_{te^+} := -1$ and $A_{te^-} := -j$. For each $e \in E$ and $u = \partial^+ e \in V \setminus S$, we put $A_{u^+e^+} = 1$ and $A_{u^-e^-} = 1$. Similarly, for each $e \in E$ and $v = \partial^- e \in V \setminus S$, we put $A_{v^+e^+} = -1$ and $A_{v^-e^-} = -1$. The other components of A are all zero.

Let $\rho : 2^E \to \mathbb{Z}$ be a function defined by $\rho(F) := \operatorname{rank} A[S \cup U^\pm, F^\pm]$ for each $F \subseteq E$, where $F^\pm := \{ e^+, e^- \mid e \in F \}$. Then F is a matching if and only if the subgraph $H = (V, F)$ is a forest with each connected component containing at most two terminals each of which belongs to different members of $\mathcal{S}$. Apparently, k disjoint $\mathcal{S}$-paths collectively form such a forest. Adding edges appropriately, one can obtain a maximal matching with k or more connected components each of which has two terminals. Conversely, for a maximal matching F, we have $|F| = |V| - c_1(F) - c_2(F)$, where $c_i(F)$ denotes the number of connected components each of which contains exactly i terminals. This together with $|S| = c_1(F) + 2c_2(F)$ implies that $|F| = |V| - |S| + c_2(F)$ holds. Since the subgraph $H = (V, F)$ includes $c_2(F)$ disjoint $\mathcal{S}$-paths, the maximum number of disjoint $\mathcal{S}$-paths can be obtained from a matching of the maximum cardinality.

In order to clarify a delta-matroid structure related to the disjoint $\mathcal{S}$-paths, we now introduce an alternating matrix M. For each $e \in E$, let τ_e denote a transcendental indeterminate, and consider a 2×2 alternating matrix

$$D_e := \begin{bmatrix} 0 & -\tau_e \\ \tau_e & 0 \end{bmatrix}$$

whose row/column set is indexed by e^+ and e^-. Let D be the block-diagonal matrix D whose diagonal blocks are D_e for all $e \in E$. Thus the row/column set of D is indexed $E^\pm$. The alternating matrix M is defined by

$$M := \begin{pmatrix} O & -A \\ A^\top & D \end{pmatrix},$$

where $A^\top$ denotes the transpose of A. The row/column set of M is indexed by $W := S \cup U^\pm \cup E^\pm$.

For a family $\mathcal{P}$ of disjoint $\mathcal{S}$-paths, we denote by $T(\mathcal{P})$ the set of terminals that appear as ends of $\mathcal{S}$-paths in $\mathcal{P}$. The following lemma characterizes when M is nonsingular.

Lemma 4.1 *The alternating matrix M is nonsingular if and only if there exists a family $\mathcal{P}$ of disjoint $\mathcal{S}$-paths such that $T(\mathcal{P}) = S$.*

Proof: Split M into $\widehat{A}$ and $\widehat{D}$ such that

$$M = \widehat{A} + \widehat{D}, \quad \widehat{A} = \begin{pmatrix} O & -A \\ A^\top & O \end{pmatrix}, \quad \widehat{D} = \begin{pmatrix} O & O \\ O & D \end{pmatrix}.$$

By [17, Lemma 7.3.20], we have

$$\operatorname{Pf} M = \sum_{Z \subseteq W} \pm \operatorname{Pf} \widehat{A}[W \setminus Z] \cdot \operatorname{Pf} \widehat{D}[Z],$$

where each sign is determined by the choice of Z. Note that $\mathrm{Pf}\,\widehat{D}[Z] \neq 0$ if and only if there exists a subset $F \subseteq E$ such that $Z = \{\, e^+, e^- \mid e \in E \setminus F \,\}$. In addition, $\mathrm{Pf}\,\widehat{A}[W \setminus Z] \neq 0$ if and only if $A[S \cup U^{\pm}, E^{\pm} \setminus Z]$ is nonsingular. Thus, we have

$$\mathrm{Pf}\, M = \sum_{F} \pm \det A[S \cup U^{\pm}, F^{\pm}] \cdot \mathrm{Pf}\, D[E^{\pm} \setminus F^{\pm}],$$

where the sum is taken over all matchings $F \subseteq E$ with $\rho(F) = |S \cup U^{\pm}|$. Since each term contains a distinct set of indeterminates, no cancellation occurs in the summation. Therefore, M is nonsingular if and only if there exists a matching F with $2|F| = |S \cup U^{\pm}| = 2|V| - |S|$.

This cardinality condition means that the number of connected components of the forest $H = (V, F)$ is $|S|/2$. Since the number of terminals in each component is at most two, this is equivalent to say that each connected component contains exactly two terminals each of which belongs to different member of $\mathcal{S}$. Since such a connected component contains a unique $\mathcal{S}$-path, we may conclude that M is nonsingular if and only if there exists a family $\mathcal{P}$ of disjoint $\mathcal{S}$-paths with $T(\mathcal{P}) = S$. □

A terminal subset $X \subseteq S$ is called $\mathcal{S}$-feasible if there exists a family $\mathcal{P}$ of disjoint $\mathcal{S}$-paths such that $X = T(\mathcal{P})$. Let $\mathcal{F}_G$ denote the family of $\mathcal{S}$-feasible sets. Then we have the following corollary.

Corollary 4.2 *For a graph $G = (V, E)$ with a terminal set $S \subseteq V$ partitioned into $\mathcal{S}$, the pair $(S, \mathcal{F}_G)$ forms an even delta-matroid.*

Proof: It follows from Lemma 4.1 that $R \subseteq S$ is $\mathcal{S}$-feasible if and only if $M[R \cup U^{\pm} \cup E^{\pm}]$ is nonsingular. By Lemma 3.1, $(W, \mathcal{F}(M))$ forms an even delta-matroid. Then $(S, \mathcal{F}_G)$ is a contraction of $(W, \mathcal{F}(M))$ by $W \setminus S$. Therefore, $(S, \mathcal{F}_G)$ is an even delta-matroid. □

Let $\mathcal{I}_G$ denote the family of all the subsets of $\mathcal{S}$-feasible sets. It has been known that $(S, \mathcal{I}_G)$ forms a matroid [25, Theorem 73.5], which is called the Mader matroid. Answering a question posed by Schrijver [25, p. 1293], Pap [22] showed that each Mader matroid is a gammoid, which implies that each Mader matroid is linear. Our construction above provides an alternative linear representation. In fact, $Y \subseteq S$ is a member of $\mathcal{I}_G$ if and only if the set of column vectors of $AD^{-1}A^{\top}$ indexed by Y is linearly independent.

5 Shortest Disjoint $\mathcal{S}$-paths

We now suppose that each edge in G has a nonnegative length $\ell(e)$. A natural approach to disjoint $\mathcal{S}$-paths of minimum total length is to utilize weighted linear matroid parity algorithms. More specifically, consider minimizing the total length $\lambda(\mathcal{P})$ among all the families $\mathcal{P}$ of disjoint $\mathcal{S}$-paths with $T(\mathcal{P}) = S$.

Let A be the matrix constructed above. Recall that a family $\mathcal{P}$ of disjoint $\mathcal{S}$-paths with $T(\mathcal{P}) = S$ is contained in a matching $F \subseteq E$ with $2|F| = 2|V| - |S|$. One can find such a matching of minimum total length $\sum_{e \in F} \ell(e)$ in polynomial time by the weighted linear matroid parity algorithms [11, 23]. The obtained edge subset F, however, does not necessarily form a disjoint $\mathcal{S}$-paths. It is certainly a collection of disjoint trees each of which contains at most two terminals from different members of $\mathcal{S}$. While a subset $F^* \subseteq F$ forms a family of disjoint $\mathcal{S}$-paths, the total length of F^* may differ from that of F. One cannot guarantee that F^* attains the minimum total length.

Yamaguchi [32] overcame this difficulty by modifying the original graph, and showed that one can find a family of k disjoint $\mathcal{S}$-paths of minimum total length in polynomial time. We utilize this idea to investigate the property of the minimum total length of disjoint $\mathcal{S}$-paths with specified terminals.

Augment the graph $G = (V, E)$ by adding two distinct vertices r and s, a new edge e_r between them, and new edges e_v connecting s and v for all $v \in V \setminus S$. The resulting graph $G' = (V', E')$ has a terminal set $S' := S \cup \{r, s\}$, which is partitioned into $\mathcal{S}' := \mathcal{S} \cup \{\{r\}, \{s\}\}$. The lengths of the new edges are set to be zero. Let A' be the matrix constructed for this setting.

If A' admits a matching F' with $2|F'| = 2|V'| - |S'|$, the subgraph $H' = (V', F')$ is a forest with each connected component containing exactly two terminals each of which belongs to different members of $\mathcal{S}$.

Then there exists a family $\mathcal{P}'$ of disjoint $\mathcal{S}'$-paths with $T(\mathcal{P}') = S'$. In particular, r and s must be in the same connected component in H'. In addition, if some other component of H' contains a vertex $v \in V \setminus S$ that does not belong to the unique $\mathcal{S}$-path in it, one can add the edge e_v and remove an appropriate one to obtain another matching of the same cardinality without increasing the total length. Thus the minimum total length is achieved by a collection of disjoint $\mathcal{S}$-paths and a tree including s and r. Since all the edges incident to s has length zero, the tree can be replaced by the star at s spanning the same vertices, and the optimal value is equal to the total length of the obtained family of disjoint $\mathcal{S}$-paths.

Construct an alternating polynomial matrix

$$M'(\theta) := \begin{pmatrix} O & -A' \\ A'^{\top} & D'(\theta) \end{pmatrix},$$

where $D'(\theta)$ is a block-diagonal matrix whose diagonal block corresponding to $e \in E'$ is a 2×2 matrix

$$D_e(\theta) := \begin{bmatrix} 0 & -\tau_e \theta^{\ell(e)} \\ \tau_e \theta^{\ell(e)} & 0 \end{bmatrix}.$$

For each $\mathcal{S}$-feasible set $R \in \mathcal{F}_G$, let $\zeta(R)$ denote the minimum value of the total length $\lambda(\mathcal{P})$ among families $\mathcal{P}$ of disjoint $\mathcal{S}$-paths with $T(\mathcal{P}) = R$. We now evaluate $\zeta(R)$ in terms of $M'(\theta)$.

Lemma 5.1 *If S is $\mathcal{S}$-feasible, then*

$$\zeta(S) = \sum_{e \in E} \ell(e) - \deg \mathrm{Pf}\, M'(\theta)$$

holds. More generally, for any $\mathcal{S}$-feasible set $R \subseteq S$, we have

$$\zeta(R) = \sum_{e \in E} \ell(e) - \deg \mathrm{Pf}\, M'(\theta)[R \cup U^{\pm} \cup E^{\pm}].$$

Proof: Suppose that S is $\mathcal{S}$-feasible. By Lemma 4.1, $M'(\theta)$ is nonsingular. As shown in the proof there, we have

$$\mathrm{Pf}\, M'(\theta) = \sum_{F} \pm \det A'[S' \cup U^{\pm}, F^{\pm}] \cdot \mathrm{Pf}\, D'(\theta)[E^{\pm} \setminus F^{\pm}],$$

where the sum is taken over all matchings F with $2|F| = 2|V| - |S|$. The degree of $\mathrm{Pf}\, M'(\theta)$ is then determined by the maximum degree of $\mathrm{Pf}\, D'(\theta)[E^{\pm} \setminus F^{\pm}]$, which is equal to the sum of the lengths of $e \in E \setminus F$. Since $\zeta(S)$ equals the minimum total length of a matching F with $2|F| = 2|V| - |S|$, we have $\deg \mathrm{Pf}\, M'(\theta) = \deg \mathrm{Pf}\, D'(\theta) = \sum_{e \in E} \ell(e) - \zeta(S)$.

Applying the same argument to $M'(\theta)[R \cup U^{\pm} \cup E^{\pm}]$ for an arbitrary $\mathcal{S}$-feasible set $R \subseteq S$, we have $\deg \mathrm{Pf}\, M'(\theta)[R \cup U^{\pm} \cup E^{\pm}] = \deg \mathrm{Pf}\, D'(\theta)[R \cup U^{\pm} \cup E^{\pm}] = \sum_{e \in E} \ell(e) - \zeta(S)$. □

For each $\mathcal{S}$-feasible set $R \in \mathcal{F}_G$, we put $\omega_G(R) = \deg \mathrm{Pf}\, M'(\theta)[R \cup U^{\pm} \cup E^{\pm}]$. Then it follows from Lemma 3.2 that ω_G is a valuation on the even delta-matroid $(S, \mathcal{F}_G)$. Lemma 5.1 shows that $\zeta(R)$ is given by $\zeta(R) = \sum_{e \in E} \ell(e) - \omega_G(R)$, which implies that $-\zeta$ is a valuation on $(S, \mathcal{F}_G)$.

6 Discrete Convexity

We now turn to the setting of openly disjoint T-paths. Let $G = (V, E)$ be a graph with a terminal set $T \subseteq V$ and a nonnegative edge length function $\ell : E \to \mathbb{R}_+$. An integer vector $x \in \mathbb{Z}^T$ is feasible if there exists a family $\mathcal{P}$ of openly disjoint T-paths such that the number of T-paths in $\mathcal{P}$ incident to u coincides with $x(u)$ for every $u \in T$. Recall that $f_G(x)$ for a feasible integer vector x is defined to be the minimum value of $\lambda(\mathcal{P}) = \sum_{P \in \mathcal{P}} \sum_{e \in E(P)} \ell(e)$ among such families $\mathcal{P}$ of openly disjoint T-paths.

In order to reduce this "openly-disjoint" setting to the "disjoint $\mathcal{S}$-paths" model, we detach all the terminal vertices, i.e., split each terminal vertex $u \in T$ into $d(u)$ vertices of degree one, where $d(u)$

denotes the degree of u. Let S_u be the set of new vertices coming from $u \in T$. Put $\mathcal{S} := \{S_u\}_{u \in T}$ and $S := \bigcup_{u \in T} S_u$. Then openly disjoint T-paths correspond to disjoint $\mathcal{S}$-paths in the resulting graph $\widehat{G}$.

Let $\mathcal{F}_{\widehat{G}}$ be the family of $\mathcal{S}$-feasible sets in $\widehat{G}$. For any subset $R \subseteq S$, we determine $z_R \in \mathbb{Z}^S$ by $z_R(u) := |R \cap S_u|$ for each $u \in T$. The set J_G of feasible integer vectors for G can be expressed by $J_G = \{ z_R \mid R \in \mathcal{F}_{\widehat{G}} \}$. Given that $(S, \mathcal{F}_{\widehat{G}})$ forms an even delta-matroid by Corollary 4.2, one can easily show that J_G forms a jump system. Thus, Corollary 4.2 provides an alternative proof for Proposition 2.1. Moreover, we have the following theorem which establishes the discrete convexity of the function $f_G : J_G \to \mathbb{R}$.

Theorem 6.1 *The function $f_G : J_G \to \mathbb{R}$ is M-convex on the jump system J_G.*

Proof: For any $x, y \in J$, there exist $X, Y \in \mathcal{F}$ such that $x = z_X$ and $y = z_Y$ with $f_G(x) = \zeta(X)$ and $f_G(y) = \zeta(Y)$. For a (z_X, z_Y)-increment a, there exists a terminal $s \in X \triangle Y$ with $s \in S_u$ and $a(u) = \pm 1$. Since $-\zeta$ is a valuation on $(S, \mathcal{F})$, there exists another terminal $t \in X \triangle Y \setminus \{s\}$ such that $\zeta(X) + \zeta(Y) \geq \zeta(X \triangle \{s,t\}) + \zeta(Y \triangle \{s,t\})$. Let $v \in T$ be the terminal with $t \in S_v$, and determine $b \in \mathbb{Z}^T$ by $b(v) = \pm 1$ and $b(w) = 0$ for $w \in T \setminus \{v\}$. More precisely, $b(v) = -1$ if $t \in X \setminus Y$ and $b(v) = 1$ if $t \in Y \setminus X$. Then b is an $(x + a, y)$-increment such that $x + a + b \in J$ and $y - a - b \in J$. Moreover, $f_G(x + a + b) \leq \zeta(X \triangle \{s,t\})$ and $f_G(y - a - b) \leq \zeta(Y \triangle \{s,t\})$ hold. Therefore, we have $f_G(x) + f_G(y) \geq f_G(x + a + b) + f_G(y - a - b)$. □

The nonnegativity assumption on the edge length function is crucial for Theorem 6.1. The graph depicted in Figure 1 with $\alpha > 0$ serves as a counterexample. An integer vector $x \in \mathbb{Z}^T$ defined by $x = (x(t_1), x(t_2), x(t_3), x(t_4)) = (1, 1, 0, 0)$ is feasible and we have $f_G(x) = -2\alpha$. The integer vector $y = (0, 0, 1, 1)$ is also feasible and we have $f_G(y) = -2\alpha$. Consider an (x, y) increment a with $a(t_1) = -1$ and $a(t_j) = 0$ for $j \neq 1$. While there are three possible choices of an $(x + a, y)$-increment b such that both $x + a + b$ and $y - a - b$ are feasible, all of them satisfy $f_G(x) + f_G(y) < f_G(x + a + b) + f_G(y - a - b)$.

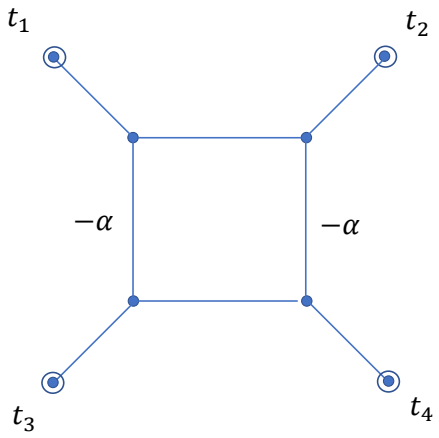


Figure 1: A counterexample to the M-convexity when some edges have negative length.

Acknowledgements

The authors are grateful to András Sebő for helpful comments on our manuscript.

References

[1] C. Berge: Sur le couplage maximum d'un graphe, *Comptes Rendu de l'Académie des Sciences*, 247 (1958), pp. 258–259.

[2] A. Bouchet: Greedy algorithm and symmetric matroids, *Math. Programming*, 38 (1987), pp. 147?159.

[3] A. Bouchet and W. H. Cunningham: Delta-matroids, jump systems, and bisubmodular polyhedra, *SIAM J. Discrete Math.*, 8 (1995), pp. 17–32.

[4] R. Chandrasekaran and S. N. Kabadi: Pseudomatroids, *Discrete Math.*, 71 (1988), pp. 205–217.

[5] B. V. Cherkassky: Solution of a problem of multiproduct flows in a network (in Russian), *Ékon. Mat. Metody*, 13 (1977), pp. 143–151.

[6] H. Y. Cheung, L. C. Lau, and K. M. Leung: Algebraic algorithms for linear matroid parity problems, *ACM Trans. Algorithms*, 10 (2014), 10: 1–26.

[7] M. Chudnovsky, W. H. Cunningham, and J. Geelen: An algorithm for packing non-zero A-paths in group-labelled graphs, *Combinatorica*, 28 (2008), pp. 145–161.

[8] A. Dress and W. Wenzel: A greedy-algorithm characterization of valuated Δ-matroids, *Appl. Math. Lett.*, 4 (1991), pp. 55–58.

[9] H. N. Gabow and M. Stallmann: An augmenting path algorithm for linear matroid parity, *Combinatorica*, 6 (1986), pp. 123–150.

[10] T. Gallai: Maximum-Minimum-Sätze und verallgemeinerte Faktoren von Graphen, *Acta Math. Acad. Sci. Hungar.*, 12 (1961), pp. 131–173.

[11] S. Iwata and Y. Kobayashi: A weighted linear matroid parity algorithm, *SIAM J. Comput.*, 51 (2022), pp. 238–280.

[12] L. Lovász: On some connectivity properties of Eulerian graphs, *Acta Math. Acad. Sci. Hungar.*, 28 (1976), pp. 129–138.

[13] L. Lovász: Matroid matching and some applications, *J. Combin. Theory*, B28 (1980), pp. 208–236.

[14] L. Lovász and M. D. Plummer, *Matching Theory*, North-Holland, Amsterdam, 1986.

[15] W. Mader: Über die Maximalzahl kantendisjunkter A-Wege, *Archiv. Math.*, 30 (1978), pp. 325–336.

[16] W. Mader: Über die Maximalzahl kreuzungsfreier H-Wege, *Archiv. Math.*, 31 (1978), pp. 387–402.

[17] K. Murota, *Matrices and Matroids for Systems Analysis*, Springer-Verlag, Berlin, 2000.

[18] K. Murota: *Discrete Convex Analysis*, SIAM, 2003.

[19] K. Murota: M-convex functions on jump systems: A general framework for minsquare graph factor problem, *SIAM J. Discrete Math.*, 20 (2006), pp. 213–226.

[20] J. B. Orlin: A fast, simpler algorithm for the matroid parity problem, *Proceedings of the 13th International Conference on Integer Programming and Combinatorial Optimization*, LNCS 5035, Springer-Verlag, 2008, pp. 240–258.

[21] J. B. Orlin and J. H. Vande Vate: Solving the linear matroid parity problem as a sequence of matroid intersection problems, *Math. Programming*, 47 (1990), pp. 81–106.

[22] G. Pap: Mader matroids are gammoids, *EGRES Technical Report*, 2006.

[23] G. Pap: Weighted linear matroid matching, *Proceedings of the Eighth Japanese-Hungarian Symposium on Discrete Mathematics and Its Applications*, 2013, pp. 411–413.

[24] A. Schrijver: A short proof of Mader's $\mathcal{S}$-paths theorem, *J. Combin. Theory*, B82 (2001), pp. 319–321.

[25] A. Schrijver: *Combinatorial Optimization — Polyhedra and Efficiency*, Springer-Verlag, 2003.

[26] M. Sadli and A. Sebő: Paths and jumps, manuscript, 2000.

[27] M. Sadli and A. Sebő: Jump-systems of T-paths, *Proceedings of the 12th Japanese-Hungarian Symposium on Discrete Mathematics and Its Applications*, 2023.

[28] A. Sebő and L. Szegő: The path-packing structure of graphs, *Proceedings of the Tenth International Conference on Integer Programming and Combinatorial Optimization*, LNCS 3064, Springer-Verlag, 2004, pp. 256–270.

[29] S. Tanigawa and Y. Yamaguchi: Packing non-zero A-paths via matroid matching, *Discrete Appl. Math.*, 214 (2016), pp. 169–178.

[30] W. T. Tutte: The factorization of linear graphs, *J. London Math.*, 22 (1947), pp. 107–111.

[31] W. Wenzel: Pfaffian forms and Δ-matroids, *Discrete Math.*, 115 (1993), pp. 253–266.

[32] Y. Yamaguchi: Shortest disjoint $\mathcal{S}$-paths via weighted linear matroid parity, *Proceedings of the 27th International Symposium on Algorithms and Computation*, 2016, 63: 1–13.

On generic universal rigidity on the line

Guilherme Zeus Dantas e Moura

Department of Mathematics and Statistics,
Haverford College, 370 Lancaster Ave,
Haverford, PA 19041, USA.
gdantasemo@haverford.edu

Tibor Jordán[1]

Department of Operations Research,
ELTE Eötvös Loránd University, and the
ELKH-ELTE Egerváry Research Group on
Combinatorial Optimization, Eötvös Loránd
Research Network (ELKH),
Pázmány Péter sétány 1/C,
1117 Budapest, Hungary.
tibor.jordan@ttk.elte.hu

Corwin Silverman

Department of Mathematics,
Grinnell College, 1115 8th Ave,
Grinnell, IA 50112, USA.
silvermanc79@gmail.com

Abstract: A d-dimensional bar-and-joint framework (G,p) with underlying graph G is called universally rigid if all realizations of G with the same edge lengths, in all dimensions, are congruent to (G,p). A graph G is said to be generically universally rigid in $\mathbb{R}^d$ if every d-dimesional generic framework (G,p) is universally rigid.

In this paper we focus on the case $d = 1$. We give counterexamples to a conjectured characterization of generically universally rigid graphs from [7]. We also introduce two new operations that preserve the universal rigidity of generic frameworks, and the property of being not universally rigid, respectively. One of these operations is used in the analysis of one of our examples, while the other operation is applied to obtain a lower bound on the size of generically universally rigid graphs. This bound gives a partial answer to a question from [11].

Keywords: rigid graph, universally rigid graph, generic framework

1 Introduction

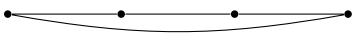
A d-dimensional (bar-and-joint) *framework* is a pair (G,p), where $G = (V,E)$ is a graph and p is a *configuration* of the vertices, that is, a map from V to $\mathbb{R}^d$. We consider the framework to be a straight line *realization* of G in $\mathbb{R}^d$. Two frameworks (G,p) and (G,q) are *equivalent* if $\|p(u)-p(v)\| = \|q(u)-q(v)\|$ holds for all pairs u,v with $uv \in E$, where $\|.\|$ denotes the Euclidean norm in $\mathbb{R}^d$. Frameworks (G,p), (G,q) are *congruent* if $\|p(u)-p(v)\| = \|q(u)-q(v)\|$ holds for all pairs u,v with $u,v \in V$. This is the same as saying that (G,q) can be obtained from (G,p) by an isometry of $\mathbb{R}^d$.

Let (G,p) be a d-dimensional framework for some $d \geq 1$. We say that (G,p) is *rigid* in $\mathbb{R}^d$ if there is a neighborhood U_p in the space of d-dimensional configurations such that if a d-dimensional framework (G,q) is equivalent to (G,p) and $q \in U_p$, then q is congruent to p. The framework (G,p) is called *globally rigid* in $\mathbb{R}^d$ if every d-dimensional framework (G,q) which is equivalent to (G,p) is congruent to (G,p). We obtain an even stronger property by extending this condition to equivalent realizations in any dimension:

[1]Research is supported in part by the Hungarian Scientific Research Fund grant no. K135421.

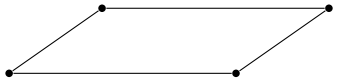
we say that (G, p) is *universally rigid* if it is a unique realization of G, up to congruence, with the given edge lengths, in all dimensions $\mathbb{R}^{d'}$, $d' \geq 1$.

Deciding whether a given framework is rigid in $\mathbb{R}^d$, for $d \geq 2$ (resp. globally rigid in $\mathbb{R}^d$, for $d \geq 1$) is NP-hard [1, 14]. The complexity of the corresponding decision problem for universal rigidity seems to be open, even for $d = 1$. These problems become more tractable, however, if we assume that there are no algebraic dependencies between the coordinates of the points of the framework. A framework (G, p) is said to be *generic* if the set containing the coordinates of all its points is algebraically independent over the rationals. It is well-known that the rigidity (resp. global rigidity) of frameworks in $\mathbb{R}^d$ is a generic property for all $d \geq 1$, that is, the (global) rigidity of (G, p) depends only on the graph G and not the particular realization p, if (G, p) is generic [2, 6, 10]. This property does not hold for universal rigidity, even if $d = 1$, which follows by considering different generic realizations of a four-cycle on the line. See Figure 1.



(a) An universally rigid realization of a four-cycle in $\mathbb{R}^1$.

(b) A not universally rigid realization of a four-cycle in $\mathbb{R}^1$.



(c) A realization of a four-cycle in $\mathbb{R}^2$ equivalent, but not congruent, to the realization in Figure 1b.

Figure 1: Realizations of a four-cycle.

A graph G is called *generically rigid* (resp. *generically globally rigid*, *generically universally rigid*) in $\mathbb{R}^d$ if every d-dimensional generic framework (G, p) is rigid (resp. globally rigid, universally rigid). Generically rigid and globally rigid graphs are well-characterized for $d \leq 2$. It remains an open problem to extend these results to $d \geq 3$. The characterization of generically universally rigid graphs is an open problem for all $d \geq 1$. We refer the reader to [12, 15] for more details on the theory of rigid graphs and frameworks.

In this paper we focus on universally rigid frameworks and generically universally rigid graphs in $\mathbb{R}^1$. We give counterexamples to a conjectured characterization of generically universally rigid graphs from [7]. We introduce a new operation that preserves the universal rigidity of generic frameworks and use it to construct infinite families of counterexamples. We also show that the so-called degree-2 extension operation preserves the property of being not universally rigid, for $d = 1$. This operation is applied in the proof of a new lower bound on the size of generically universally rigid graphs. This bound gives a partial answer to a question from [11].

2 Examples

Let $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$ be two graphs for which $V_1 \cap V_2$, $V_1 - V_2$, and $V_2 - V_1$ are all nonempty. Then $G = (V_1 \cup V_2, E_1 \cup (E_2 - E(G_2[V_1 \cap V_2])))$ is called the *edge reduced attachment* of G_1 and G_2 *along* G_2. That is, G is obtained by removing the edges of G_2 which are spanned by the intersection of their vertex sets and then taking the union of the two graphs. Ratmanski [13] proved that the edge reduced attachment operation preserves generic universal rigidity in $\mathbb{R}^d$, provided $|V_1 \cap V_2| \geq d + 1$. It was conjectured, by participants of a workshop in 2011, that for $d = 1$ every generically universally rigid graph can be obtained from a set of triangles by this operation, and edge addition.

Conjecture 1. *[7] A graph G on at least three vertices is generically universally rigid in $\mathbb{R}^1$ if and only if G can be obtained from a set of triangles by edge reduced attachment and edge addition operations.*

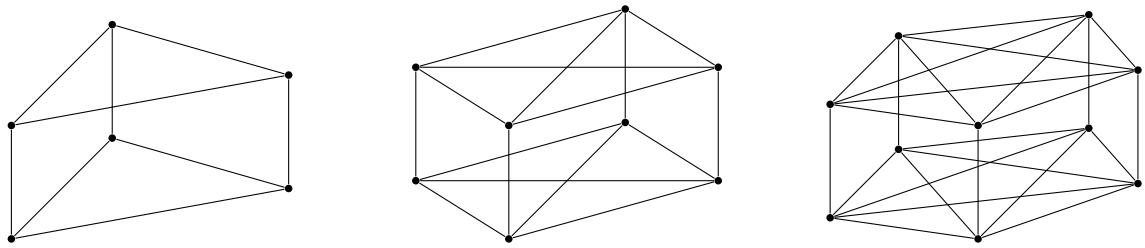
We next present two counterexamples to Conjecture 1. The first one, on eight vertices, requires a more sophisticated argument, including the analysis of a new operation that can be used to build generically

universally rigid graphs. This operation can also be used to construct infinite families of counterexamples. The second one, on sixteen vertices, is fairly easy to verify. We need the following simple lemma on attachments.

The K_4-*completion* operation adds a new edge uv to a graph for a vertex pair u, v with two adjacent common neighbours. We say that $F \subseteq E$ is an *independent edge cut* in a graph $G = (V, E)$ if the edges in F are pairwise disjoint, and there is a nonempty proper subset $X \subset V$ for which the set of edges connecting X and $V - X$ is G is F.

Lemma 2. *Suppose that $G = (V, E)$ is a connected graph that can be obtained by edge reduced attachments and edge additions from a set of triangles. Then*
(i) G contains a triangle,
(ii) the complete graph on V can be obtained from G by K_4-completion operations,
(iii) G has no independent edge cuts.

Proof: (i) follows, by induction, from the fact that if G is the edge reduced attachment of G_1 and G_2 along G_2, then G contains G_1 as a subgraph. So if G_1 contains a triangle, so does G. To prove (ii) we use induction on the number t of operations used to build up G. For $t = 0$ we have $G = K_3$, for which the statement is obvious. Suppose that $t \geq 1$ and consider the last operation applied, that resulted in graph G. The case when the last operation is edge addition is easy to deal with, so suppose that G was obtained from G_1 and G_2 by an edge reduced attachment along G_2. Then G_1 is a subgraph of G. By induction the complete graph on $V(G_1)$ can be obtained from G_1 by K_4-completions. By performing these operations on G, we make $V(G_1) \cap V(G_2)$ complete, which implies that G_2 is a subgraph of the resulting graph. So, by induction, we can apply K_4-completions to make the subgraph of G on $V(G_2)$ complete as well. Since G_1 and G_2 share at least two vertices, all the missing edges of G can then be added by further K_4-completions. Finally, it is clear that (ii) implies (iii). □



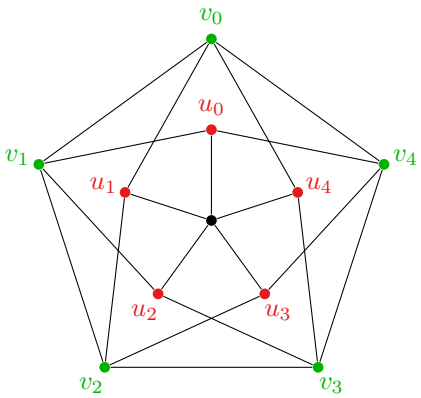
(a) The graph B_3. (b) The graph B_4. (c) The graph B_5.

Figure 2: Graphs B_3, B_4, and B_5.

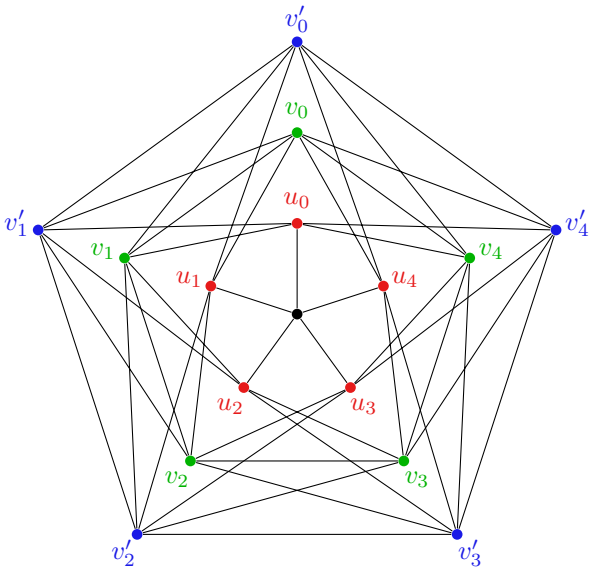
Let B_n be the graph obtained from two disjoint complete graphs on n vertices by adding n disjoint edges. See Figure 2. As we shall see in the next section (c.f. Theorem 7), B_n is generically universally rigid in $\mathbb{R}^1$ for $n \geq 4$. Since B_n has an independent edge cut, it cannot be obtained by edge reduced attachments and edge additions from a set of triangles by Lemma 2(iii). Hence B_4 (and each graph B_i, for $i \geq 4$) is a counterexample to Conjecture 1.

The other example was motivated by a question in [11], asking whether there is a triangle-free generically universally rigid graph in $\mathbb{R}^1$, and in particular, whether the triangle-free Grötzsch graph is generically universally rigid in $\mathbb{R}^1$. See Figure 3a. We leave the latter question open. Instead we consider the following supergraph of the Grötzsch graph and use it to give an affirmative answer to the former question.

Definition 3. *The augmented Grötzsch graph is obtained from the Grötzsch graph on vertex set $\{w, u_0, u_1, \ldots u_4, v_0, v_1 \ldots v_4\}$ (labeled as in Figure 3a) by adding the vertices v_i', $0 \leq i \leq 4$ and edges $v_i'u_{i+1}, v_i'v_{i+1}, v_i'v_{i+1}'$, for $0 \leq i \leq 4$, counting indices mod 5. See Figure 3b.*



(a) The Grötzsch graph.



(b) The augmented Grötzsch graph.

Figure 3: The Grötzsch graph and the augmented Grötzsch graph. The central vertex is w.

Thus the augmented Grötzsch graph has sixteen vertices, and it contains the Grötzsch graph as a subgraph. We shall prove that it is generically universally rigid in $\mathbb{R}^1$. We need the following simple lemma. The *degree-2 extension* operation adds a new vertex w to a graph G and two new edges wx, wy, for two distinct vertices of G. This operation can also be performed on a framework (G, p), in which case it includes the extension of p by $p(w)$.

Lemma 4. *Let (G, p) be a universally rigid realization of G in $\mathbb{R}^1$. Suppose that (G', p) can be obtained from (G, p) by a degree-2 extension that adds the edges wx, wy. If $p(x) \neq p(y)$, then (G', p) is universally rigid in $\mathbb{R}^1$.*

Let C be a cycle on vertex set $\{x_1, x_2, \ldots, x_k\}$ with edge set $E(C) = \{x_1x_2, \ldots, x_{k-1}x_k, x_kx_1\}$. Let (C, p) be a 1-dimensional realization of C. If $p(x_1) < p(x_2) < \cdots < p(x_k)$ then (C, p) (or sometimes C itself) is called a *stretched cycle*. It is easy to see that stretched cycles are universally rigid in $\mathbb{R}^1$.

It is known that the Grötzsch graph is not a "cover graph." By using our terminology, this fact can be restated as follows. See [8] for a short combinatorial proof.

Theorem 5. *[8] Every injective 1-dimensional realization of the Grötzsch graph has a stretched cycle.*

We are ready to deduce that the augmented Grötzsch graph is generically universally rigid. We can actually show a somewhat stronger property.

Theorem 6. *Every injective 1-dimensional realization of the augmented Grötzsch graph is universally rigid.*

Proof: Let G be the augmented Grötzsch graph and let G' denote its subgraph isomorphic to the Grötzsch graph, obtained by deleting the vertices v'_i, $0 \leq i \leq 4$. Consider an injective 1-dimensional realization (G, p). By Theorem 5 there is a stretched cycle C on vertices $\{x_1, x_2, \ldots, x_k\}$ in $(G', p|_{V(G')})$. Since G' is triangle-free, we must have $k \geq 4$. Let $(H, p|_{V(H)})$ be a maximal universally rigid subframework of (G, p) with $V(C) \subseteq V(H)$. Such a framework exists, since the subframework of the stretched cycle C is universally rigid. We shall prove that $H = G$.

The structure of G' and the fact that C contains a path on three vertices which is disjoint from w implies that we must have two vertices in $C - w$ with identical indices or two vertices whose indices differ by two. Formally, either (1) there exists a pair of vertices u_i, v_i in C, or (2) there exists a pair of vertices u_i, v_{i+2} (or u_i, u_{i+2}, v_i, v_{i+2}, v_i, u_{i+2}), for some $0 \leq i \leq 4$.

Let us consider case (1). By symmetry we may assume that $u_1, v_1 \in V(C)$. Then, since u_1, v_1 are both neighbours of v_2 and v_2', Lemma 4 implies that $v_2, v_2' \in V(H)$ holds. We can apply a similar argument to v_3, v_3', and so on around the cycle, to deduce that H contains all v- and v'-vertices. Finally, the u-vertices and vertex w can also be added by degree-2 extensions. Thus $H = G$ and the theorem follows.

Next consider case (2). Suppose that, say, $u_1, v_3 \in V(C)$. Then, since u_1, v_3 are both neighbours of v_2 and v_2', Lemma 4 implies that $v_2, v_2' \in V(H)$ holds. The rest of the argument is identical to that of case (1). This completes the proof. □

Since the augmented Grötzsch graph is triangle-free and, by Theorem 6, generically universally rigid in $\mathbb{R}^1$, it follows from Lemma 2(i) that it is a counterexample to Conjecture 1.

It may be interesting to find graphs with arbitrarily large girth which are generically universally rigid in $\mathbb{R}^1$ (or possibly in higher dimensions).

3 Operations

3.1 Combining graphs along disjoint edges

If G has an independent edge cut J, then the removal of J from G results in two smaller (sub)graphs. The reversal of this operations can be defined as follows. Let $G = (V, E)$, $H = (U, F)$ be two disjoint graphs. Let $v_1, v_2, \ldots, v_k \in V$ and $u_1, u_2, \ldots, u_k \in U$ be distinct vertices. The *join* $G \sqcup H$ of G and H is the graph on vertex set $V \cup U$ with edge set $E \cup F \cup \{v_i u_i : i \in \{1, 2, \ldots, k\}\}$. We say that $G \sqcup H$ is obtained from G and H by a join operation *along k edges*. See Figure 4. For example, B_n is the join of two complete graphs on n vertices along n edges.

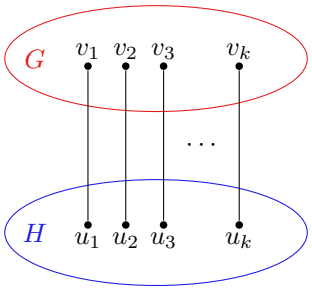


Figure 4: The graph obtained from G and H by a join operation along k edges.

Theorem 7. *Let G and H be generically universally rigid graphs in $\mathbb{R}^1$ on at least k vertices and let $G \sqcup H$ be obtained from G and H by a join operation along k edges. If $k \geq 4$, then $G \sqcup H$ is generically universally rigid in $\mathbb{R}^1$.*

Proof: Let $G = (V, E)$, $H = (U, F)$, $B = G \sqcup H$. We may assume that $k = 4$. Consider a 1-dimensional generic realization (B, p). Suppose that (B, q) is an equivalent realization in $\mathbb{R}^d$, for some $d \geq 1$. We shall prove that p and q are congruent.

Note that $(G, p|_V)$ is a generic 1-dimensional realization of G, and $(G, q|_V)$ is an equivalent realization. Hence, since G is generically universally rigid in $\mathbb{R}^1$, it follows that $(G, p|_V)$ is congruent to $(G, q|_V)$. Analogously, we also have that $(H, p|_U)$ is congruent to $(H, q|_U)$. Hence, there exists $\mathbf{c}_G, \mathbf{d}_G, \mathbf{c}_H, \mathbf{d}_H \in \mathbb{R}^d$,

with $\|\mathbf{d}_G\| = \|\mathbf{d}_H\| = 1$, such that

$$q(v) = \mathbf{c}_G + p(v)\mathbf{d}_G \qquad \forall v \in V, \qquad \text{and} \tag{1}$$
$$q(u) = \mathbf{c}_H + p(u)\mathbf{d}_H \qquad \forall u \in U. \tag{2}$$

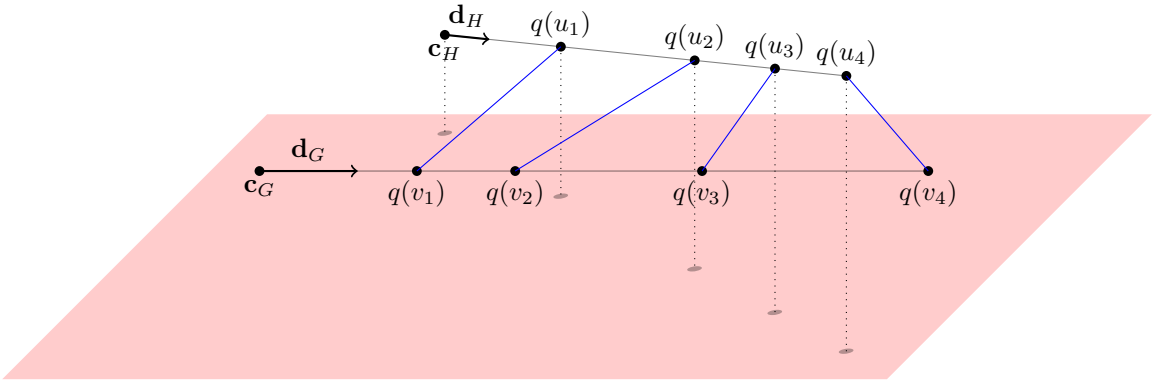


Figure 5

Since p and q are equivalent, for each $i \in \{1, 2, 3, 4\}$, we have

$$\|q(v_i) - q(u_i)\|^2 = \|p(v_i) - p(u_i)\|^2. \tag{3}$$

By changing the square of norms to dot product we obtain that, for each $i \in \{1, 2, 3, 4\}$,

$$\Big((\mathbf{c}_G + p(v_i)\mathbf{d}_G) - (\mathbf{c}_H + p(u_i)\mathbf{d}_H)\Big) \cdot \Big((\mathbf{c}_G + p(v_i)\mathbf{d}_G) - (\mathbf{c}_H + p(u_i)\mathbf{d}_H)\Big) - (p(v_i) - p(u_i))^2 = 0. \tag{4}$$

This gives, by using $\|\mathbf{d}_G\| = \|\mathbf{d}_H\| = 1$, that for each $i \in \{1, 2, 3, 4\}$,

$$\begin{aligned} &\mathbf{c}_G \cdot \mathbf{c}_G - 2\mathbf{c}_G \cdot \mathbf{c}_H + \mathbf{c}_H \cdot \mathbf{c}_H + \\ &2(1 - \mathbf{d}_G \cdot \mathbf{d}_H)\, p(u_i)p(v_i) + \\ &2(\mathbf{c}_H \cdot \mathbf{d}_H - \mathbf{c}_G \cdot \mathbf{d}_H)p(u_i) + \\ &2(\mathbf{c}_G \cdot \mathbf{d}_G - \mathbf{c}_H \cdot \mathbf{d}_G)p(v_i) = 0. \end{aligned} \tag{5}$$

By applying equation (5) for $i \in \{1, 2, 3\}$, as well as for $i = 4$, and by subtracting, we obtain that for each $i \in \{1, 2, 3\}$,

$$\begin{aligned} &2(1 - \mathbf{d}_G \cdot \mathbf{d}_H)\,(p(u_i)p(v_i) - p(u_4)p(v_4)) + \\ &2(\mathbf{c}_H \cdot \mathbf{d}_H - \mathbf{c}_G \cdot \mathbf{d}_H)(p(u_i) - p(u_4)) + \\ &2(\mathbf{c}_G \cdot \mathbf{d}_G - \mathbf{c}_H \cdot \mathbf{d}_G)(p(v_i) - p(v_4)) = 0. \end{aligned} \tag{6}$$

Let us define $f_1(i) = p(u_i)p(v_i) - p(u_4)p(v_4)$, $f_2(i) = p(u_i) - p(u_4)$, and $f_3(i) = p(v_i) - p(v_4)$. Then we have, for each $i \in \{1, 2, 3\}$,

$$2(1 - \mathbf{d}_G \cdot \mathbf{d}_H)f_1(i) + 2(\mathbf{c}_H \cdot \mathbf{d}_H - \mathbf{c}_G \cdot \mathbf{d}_H)f_2(i) + 2(\mathbf{c}_G \cdot \mathbf{d}_G - \mathbf{c}_H \cdot \mathbf{d}_G)f_3(i) = 0. \tag{7}$$

By applying equation (7) for $i \in \{1, 2\}$, as well as for $i = 3$, and by multiplying the respective equations by $f_3(3)$ and $f_3(i)$, and then subtracting, we obtain that for each $i \in \{1, 2\}$,

$$2(1 - \mathbf{d}_G \cdot \mathbf{d}_H)(f_1(i)f_3(3) - f_1(3)f_3(i)) + 2(\mathbf{c}_H \cdot \mathbf{d}_H - \mathbf{c}_G \cdot \mathbf{d}_H)(f_2(i)f_3(3) - f_2(3)f_3(i)) = 0 \tag{8}$$

Let us define $f_4(i) = f_1(i)f_3(3) - f_1(3)f_3(i)$, and $f_5(i) = f_2(i)f_3(3) - f_2(3)f_3(i)$. Then we have

$$2(1 - \mathbf{d}_G \cdot \mathbf{d}_H)f_4(1) + 2(\mathbf{c}_H \cdot \mathbf{d}_H - \mathbf{c}_G \cdot \mathbf{d}_H)f_5(1) = 0, \tag{9}$$
$$2(1 - \mathbf{d}_G \cdot \mathbf{d}_H)f_4(2) + 2(\mathbf{c}_H \cdot \mathbf{d}_H - \mathbf{c}_G \cdot \mathbf{d}_H)f_5(2) = 0. \tag{10}$$

Let us define $f = f_4(1)f_5(2) - f_4(2)f_5(1)$. By multiplying equation (9) by $f_5(2)$, equation (10) by $f_5(1)$, and taking the difference, we obtain

$$2(1 - \mathbf{d}_G \cdot \mathbf{d}_H)f = 0. \tag{11}$$

Therefore, $f = 0$ or $\mathbf{d}_G \cdot \mathbf{d}_H = 1$ must hold. In the former case, f is a non-zero rational polynomial in eight variables with $p(v_1)$, $p(v_2)$, $p(v_3)$, $p(v_4)$, $p(u_1)$, $p(u_2)$, $p(u_3)$, and $p(u_4)$ as a root, which contradicts the genericity of p.

The explicit form of the polynomial is as follows:

$$\begin{aligned}
f = &-p(u_1)p(u_3)p(v_1)p(v_2)p(v_3) + p(u_2)p(u_3)p(v_1)p(v_2)p(v_3) + p(u_1)p(u_4)p(v_1)p(v_2)p(v_3)\\
&-p(u_2)p(u_4)p(v_1)p(v_2)p(v_3) + p(u_1)p(u_2)p(v_1)p(v_3)^2 - p(u_2)p(u_3)p(v_1)p(v_3)^2\\
&-p(u_1)p(u_4)p(v_1)p(v_3)^2 + p(u_3)p(u_4)p(v_1)p(v_3)^2 - p(u_1)p(u_2)p(v_2)p(v_3)^2\\
&+p(u_1)p(u_3)p(v_2)p(v_3)^2 + p(u_2)p(u_4)p(v_2)p(v_3)^2 - p(u_3)p(u_4)p(v_2)p(v_3)^2\\
&+p(u_1)p(u_3)p(v_1)p(v_2)p(v_4) - p(u_2)p(u_3)p(v_1)p(v_2)p(v_4) - p(u_1)p(u_4)p(v_1)p(v_2)p(v_4)\\
&+p(u_2)p(u_4)p(v_1)p(v_2)p(v_4) - 2\,p(u_1)p(u_2)p(v_1)p(v_3)p(v_4) + p(u_1)p(u_3)p(v_1)p(v_3)p(v_4)\\
&+p(u_2)p(u_3)p(v_1)p(v_3)p(v_4) + p(u_1)p(u_4)p(v_1)p(v_3)p(v_4) + p(u_2)p(u_4)p(v_1)p(v_3)p(v_4)\\
&-2\,p(u_3)p(u_4)p(v_1)p(v_3)p(v_4) + 2\,p(u_1)p(u_2)p(v_2)p(v_3)p(v_4) - p(u_1)p(u_3)p(v_2)p(v_3)p(v_4)\\
&-p(u_2)p(u_3)p(v_2)p(v_3)p(v_4) - p(u_1)p(u_4)p(v_2)p(v_3)p(v_4) - p(u_2)p(u_4)p(v_2)p(v_3)p(v_4)\\
&+2\,p(u_3)p(u_4)p(v_2)p(v_3)p(v_4) - p(u_1)p(u_3)p(v_3)^2p(v_4) + p(u_2)p(u_3)p(v_3)^2p(v_4)\\
&+p(u_1)p(u_4)p(v_3)^2p(v_4) - p(u_2)p(u_4)p(v_3)^2p(v_4) + p(u_1)p(u_2)p(v_1)p(v_4)^2\\
&-p(u_1)p(u_3)p(v_1)p(v_4)^2 - p(u_2)p(u_4)p(v_1)p(v_4)^2 + p(u_3)p(u_4)p(v_1)p(v_4)^2\\
&-p(u_1)p(u_2)p(v_2)p(v_4)^2 + p(u_2)p(u_3)p(v_2)p(v_4)^2 + p(u_1)p(u_4)p(v_2)p(v_4)^2\\
&-p(u_3)p(u_4)p(v_2)p(v_4)^2 + p(u_1)p(u_3)p(v_3)p(v_4)^2 - p(u_2)p(u_3)p(v_3)p(v_4)^2\\
&-p(u_1)p(u_4)p(v_3)p(v_4)^2 + p(u_2)p(u_4)p(v_3)p(v_4)^2.
\end{aligned} \tag{12}$$

In the latter case, $\mathbf{d}_G \cdot \mathbf{d}_H = 1$ (along with $\|\mathbf{d}_G\| = \|\mathbf{d}_H\| = 1$) imply that $\mathbf{d}_G = \mathbf{d}_H$. Let $\mathbf{d} = \mathbf{d}_G = \mathbf{d}_H$ and $\mathbf{t} = \mathbf{c}_H - \mathbf{c}_G$.

Suppose that $\mathbf{t} \neq \mathbf{0}$. By applying an isometry of $\mathbb{R}^d$ to (B, q) we may suppose that $p|_V = q|_V$. This implies that $(H, p|_U)$ and $(H, q|_U)$ are two congruent realizations of H on two parallel lines L_p and L_q, with the same vertex ordering. Furthermore, L_p is the line that contains (B, p). Then $q(u_i) - p(u_i)$ are parallel for $1 \leq i \leq 4$. Since (B, p) and (B, q) are equivalent, we have $||q(v_i) - q(u_i)|| = ||p(v_i) - p(u_i)||$ for $i = 1, 2$. But this gives $||p(v_1) - p(u_1)|| = ||p(v_2) - p(u_2)||$, contradicting the genericity of p.

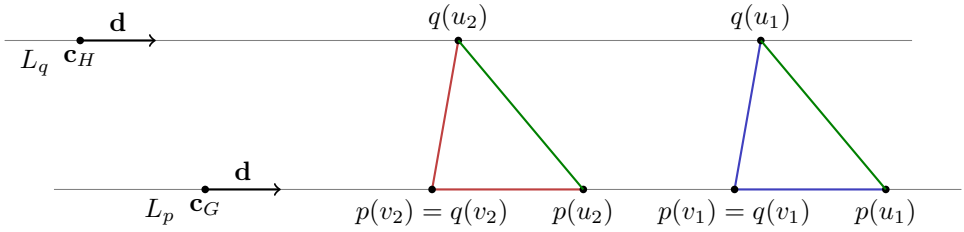


Figure 6: Diagram for the case $\mathbf{d}_G = \mathbf{d}_H$ and $\mathbf{c}_G \neq \mathbf{c}_H$. The green segments are parallel, the red segments have equal lengths, and the blue segments have equal lengths. It must be the case that the red and blue lengths are equal.

Hence $\mathbf{t} = \mathbf{0}$, which gives $\mathbf{d}_G = \mathbf{d}_H$ and $\mathbf{c}_G = \mathbf{c}_H$. Therefore p is congruent to q, as desired. Thus B is generically universally rigid in $\mathbb{R}^1$. $\square$

We can use Theorem 7 and the results of the previous section to construct an infinite family of triangle free generically universally rigid graphs on the line.

Theorem 7 does not hold for $k \leq 3$. For example, the join of two K_3 graphs along three edges (the so-called prism, or Desargue graph) is not generically universally rigid in $\mathbb{R}^1$. For a detailed analysis of this graph see [4].

3.2 Degree-2 extension

Let (G,p) be a framework on the line with $G=(V,E)$. A pair of vertices $\{u,v\}$, $u,v \in V$ is called *universally linked* in (G,p) if $\|q(u)-q(v)\| = \|p(u)-p(v)\|$ holds for all frameworks (G,q) which are equivalent to (G,p) (in all dimensions).

We believe that if $\{u,v\}$ is not universally linked in (H,p), and (G,p) is obtained from (H,p) by a degree-2 extension, then $\{u,v\}$ is not universally linked in (G,p). We can prove the following somewhat weaker statement.

Theorem 8. *Suppose that H is not generically universally rigid. Let G be obtained from H by a degree-2 extension. Then G is not generically universally rigid.*

Proof: We may assume that $|V(H)| \geq 4$, since the statement is easy to verify when H has at most three vertices. By our assumption H is not generically universally rigid, hence there exists a generic 1-dimensional realization (H,p) and a pair $\{u,v\}$ of vertices of H which is not universally linked in (H,p). Thus there exists an equivalent framework (H,q), such that $|p(u)-p(v)| \neq |q(u)-q(v)|$. Let G be obtained from H by adding a vertex w and edges wx, wy. Let $\alpha = |p(x)-p(y)|$ and $\beta = |q(x)-q(y)|$. Since p is generic, we have $\alpha \neq 0$.

Suppose first that we have $\beta \geq \alpha$. Then we can choose a point r on the line (in the complement of the line segment connecting $p(x)$ and $p(y)$), for which the set $r \cup \{p(z) : z \in V(H)\}$ is generic, and $|r-p(x)| + |r-p(y)| \geq \beta$. Since we also have $||r-p(x)| - |r-p(y)|| = \alpha \leq \beta$, we can find a point s (in a plane that contains $q(x), q(y)$, and third point of (H,q)) for which $|s-q(x)| = |r-p(x)|$ and $|s-q(y)| = |r-p(y)|$ holds. Then by adding w to (H,p) so that $p(w)=r$, and adding w to (H,q) so that $q(w)=s$ we obtain a pair of equivalent realizations of G for which (G,p) is generic and $\{u,v\}$ is not universally linked in (G,p).

Next suppose that $\alpha > \beta$. If $\beta \neq 0$ then we can use a similar argument as follows. We choose a point r on the line (in the interior of the line segment connecting $p(x)$ and $p(y)$), for which the set $r \cup \{p(z) : z \in V(H)\}$ is generic, and $||r-p(x)| - |r-p(y)|| \leq \beta$. Since we also have $|r-p(x)| + |r-p(y)| = \alpha > \beta$, we can find a point s (in a plane that contains $q(x), q(y)$, and third point of (H,q)) for which $|s-q(x)| = |r-p(x)|$ and $|s-q(y)| = |r-p(y)|$ holds. Then by adding w to (H,p) so that $p(w)=r$, and adding w to (H,q) so that $q(w)=s$ we obtain a pair of equivalent realizations of G for which (G,p) is generic and $\{u,v\}$ is not universally linked in (G,p).

Finally, we consider the case when $\beta = 0$, that is, $q(x)$ and $q(y)$ are coincident. Then we modify (H,q) by applying a result of Bezdek and Connelly [3], which states that there is a continuous motion $\Phi : [0,1] \to \mathbb{R}^{2d|V|}$ from (H,p) to (H,q) in $\mathbb{R}^{2d}$, where d is the affine dimension of (H,q). Moreover, the distances between all pairs of vertices change in a monotone way during the motion. If there is a realization (H,q') on the trajectory of this motion for which $q'(x) \neq q'(y)$ and $|p(u)-p(v)| \neq |q'(u)-q'(v)|$, then the theorem follows by applying the above arguments to (H,q') in place of (H,q).

If there is no such realization then x and y becomes coincident (at time $t \in [0,1]$, for some $0 < t < 1$) before the distance between u and v begins to change. However, there must be another pair of non-adjacent vertices $\{u',v'\}$ for which the trajectory in $[0,t)$ contains a framework (H,q') with $q'(x) \neq q'(y)$ and $|p(u')-p(v')| \neq |q'(u')-q'(v')|$. This follows from the fact that the graph $K_n - e$, obtained from a complete graph by deleting an edge, is generically universally rigid in $\mathbb{R}^1$, for $n \geq 4$ (which is easy to verify). So it is not possible that all but one of the pairwise distances stay the same in, say $[0, \frac{t}{2}]$. The theorem then follows by applying the above arguments to (H,q') in place of (H,q) and $\{u',v'\}$ in place of $\{u,v\}$. $\square$

Note that Theorem 8 does not hold if we replace degree-2 extension by degree-3 extension: consider $K_4 - e$ and remove a vertex of degree three.

4 A lower bound on the number of edges

The following question was posed in [11].

Question 9. *[11, Question 3.6] Let $G = (V, E)$ be generically universally rigid in $\mathbb{R}^1$. Does this imply that $|E| \geq 2|V| - 3$?*

If we replace universally rigid by globally rigid (resp. rigid), then the best possible lower bound is $|V|$ (resp. $|V| - 1$), which is attained by the family of cycles (resp. trees). As an application of Theorem 8 we show that generically universally rigid graphs need more edges indeed, namely, at least $\frac{3}{2}|V|$ (assuming $|V| \geq 6$). This bound is the first step towards an affirmative answer to Question 9. Note that $2|V| - 3$ would be best possible, as shown by the graphs $K_{|V|-2,2} + e$, obtained from the complete bipartite graph $K_{|V|-2,2}$ by adding an edge.

Theorem 10. *Let $G = (V, E)$ be a graph with $|V| \geq 6$ and $|E| < \frac{3}{2}|V|$. Then G is not generically universally rigid in $\mathbb{R}^1$.*

Proof: By induction on $|V|$. For $|V| = 6$ it is easy to see that no graph with at most eight edges is generically universally rigid: suppose H is such a graph. Then H must have a vertex v of degree two. By Theorem 8 $H - v$ is also generically universally rigid. It also has a vertex w of degree two. Then $H - \{v, w\}$ has four vertices, at most four edges, and, again by Theorem 8, is generically universally rigid, which is impossible.

Consider the inductive step and let $|V| \geq 7$. Let us suppose, for a contradiction, that G is generically universally rigid. The edge count and the fact that generically universally rigid graphs on at least three vertices have minimum degree at least two implies that G has a vertex v of degree two. By Theorem 8 $G - v$ is generically universally rigid. It implies, by induction, that $G - v$ has at least $\frac{3}{2}(|V| - 1)$ edges. This gives $|E| \geq |E(G - v)| + 2 \geq \frac{3}{2}(|V| - 1) + 2 \geq \frac{3}{2}|V|$, a contradiction. □

5 Concluding remarks

Instead of the genericity assumption on p, we may consider frameworks whose vertices are in general position, or frameworks for which p is injective, or quasi-injective. (Quasi-injective means that the endvertices of an edge cannot be coincident.) These properties can also be used to define families of graphs G by requiring that some (or every) 1-dimensional general position (resp. injective, quasi-injective) realization of G is universally rigid. The characterization of these families, in most cases, is still open. See [9, Table 3] for a good overview.

To motivate further research in this direction, we show that the bound in Question 9 above is tight if we replace generic by quasi-injective.

Theorem 11. *Suppose that every quasi-injective realization of $G = (V, E)$ in $\mathbb{R}^1$ is universally rigid. Then $|E| \geq 2|V| - 3$.*

Proof: Let us assume that $|E| \leq 2|V| - 4$. By a result of Chen and Yu [5] this implies that G has an independent vertex separator, that is, a set $S \subset V$ for which $G - S$ is disconnected and G has no edges with both endvertices in S. We can use this fact to construct a quasi-injective 1-dimensional realization (G, p) which is not universally rigid (in fact not even globally rigid): we choose a quasi-injective map p so that all vertices of S are mapped to the same point s on the line. Then an equivalent, non-congruent realization (G, q) can be obtained by rotating the vertices of some connected component of $G - S$ about s. □

Since every quasi-injective realization of the graphs $K_{|V|-2,2} + e$ on the line is universally rigid by Lemma 4, the bound in Theorem 11 is tight.

Acknowledgements

The results of this paper were obtained in the framework of an undergraduate research experience project of BSM (Budapest Semesters in Mathematics), led by the second author. We thank Owen Cardwell and Keegan Stump for some useful comments.

References

[1] T.G. Abbott, *Generalizations of Kempe's universality theorem*, Master's thesis, Massachusetts Institute of Technology, Dept. of Electrical Engineering and Computer Science, 2008.

[2] L. Asimow and B. Roth, *Rigidity of graphs II*, J. Math. Anal. Appl. **68** (1979) 171–190.

[3] K. Bezdek and R. Connelly, Pushing disks apart - the Kneser-Poulsen conjecture in the plane, *J. reine angew. Math.* 553 (2002), 221-236.

[4] B. Chen, R. Connelly, S.J. Gortler, A. Nixon, and L. Theran, Universal rigidity of ladders on the line, arXiv:2207.08763, 2022.

[5] G. Chen and X. Yu, A note on fragile graphs, *Discrete Mathematics* 249 (2002) 41-43.

[6] R. Connelly, *Generic global rigidity*, Discrete Comput. Geom. **33** (2005), 549–563.

[7] R. Connelly (ed.), *Conjectures and questions on global rigidity*, Open problems of a mini-workshop at Cornell University, February 2011.

[8] D.C. Fisher, K. Fraughnaugh, L. Langley, and D.B. West, The number of dependent arcs in an acyclic orientation, *J. Combin. Theory*, Ser. B **71** (1997), 73–78.

[9] D. Garamvölgyi, Global rigidity of (quasi-)injective frameworks on the line, *Discrete Mathematics* Vol. 345, Issue 2, February 2022.

[10] S. Gortler, A. Healy, and D. Thurston, *Characterizing generic global rigidity*, Amer. J. Math. **132**(4) (2010), 897–939.

[11] T. Jordán and V-H. Nguyen, On universally rigid frameworks on the line, *Cont. Disc. Math.*, vol. 10, no. 2, pp. 10-21, 2015.

[12] T. Jordán and W. Whiteley, Global rigidity, *in* J.E. Goodman, J. O'Rourke, C.D. Tóth (eds.), *Handbook of Discrete and Computational Geometry,* 3rd ed., CRC Press, Boca Raton, 2018.

[13] K. Ratmanski, Universally rigid framework attachments, arXiv:1011.4094, 2010.

[14] J.B. Saxe, *Embeddability of weighted graphs in k-space is strongly NP-hard.* Technical report, Computer Science Department, Carnegie Mellon University (1979).

[15] B. Schulze and W. Whiteley, Rigidity and scene analysis, *in* J.E. Goodman, J. O'Rourke, C.D. Tóth (eds.), *Handbook of Discrete and Computational Geometry,* 3rd ed., CRC Press, Boca Raton, 2018.

Radon number of graph families

Attila Jung[1]

Department of Computer Science
Eötvös Loránd University
Budapest, Hungary
jungattila@gmail.com

Abstract: Motivated by Bukh's counterexample to Eckhoff's Partition Conjecture, we define Radon numbers for families of isomorphism classes of graphs and completely characterize families with Radon number at most four in terms of small forbidden subgraphs.

Keywords: convexity spaces, Radon numbers, Eckhoff's conjecture, graphs

1 Introduction

Radon's Lemma [6] states that whenever we have $X \subset \mathbb{R}^d$ with $|X| \geq d+2$, we can partition X into two disjoint subsets $X^+ \cup X^- = X$ such that $\text{conv}(X^+) \cap \text{conv}(X^-) \neq \emptyset$, where $\text{conv}(S)$ is the convex hull of $S \subset \mathbb{R}^d$. One of its generalizations is Tverberg's Theorem [7], which can be stated as the following. For any $k \geq 2$ and $X \subset \mathbb{R}^d$, if we have $|X| \geq (d+1)(k-1)+1$, then we can partition X into disjoint subsets $X_1 \cup \ldots \cup X_k$ such that $\bigcap_i \text{conv}(X_i) \neq \emptyset$. Eckhoff's Partition Conjecture [2, 3] states, that Tverberg's Theorem is a purely combinatorial consequence of Radon's Lemma. For an exact statement, we need to define abstract convexity spaces and their generalized Radon numbers. For an overview of convexity spaces and their invariants see the book by van de Vel [8].

Definition 1 *Let X be a set and let $\mathcal{C}$ be a family of subsets of X. The pair $(X, \mathcal{C})$ is a convexity space, if*

- $\emptyset, X \in \mathcal{C}$,
- *If $\mathcal{C}' \subset \mathcal{C}$, then $\bigcap_{C \in \mathcal{C}'} C \in \mathcal{C}$,*
- *If $C_1, C_2, \ldots \in \mathcal{C}$ and $C_1 \subseteq C_2 \subseteq ....$, then $\cup_i C_i \in \mathcal{C}$.*

In this sense $\mathbb{R}^d$ and the family of all the convex sets in $\mathbb{R}^d$ forms a convexity space. We can define the convex hull of a subset $S \subseteq X$ in a convexity space as

$$\text{conv}(S) = \bigcap_{S \subset C \in \mathcal{C}} C.$$

Definition 2 *The kth generalized Radon number of a convexity space $(X, \mathcal{C})$ is the smallest number r_k such that any subset $S \subseteq X$ with $|S| = r_k$ can be partitioned into k nonempty disjoint subsets $S = \cup_{i=1}^k S_i$ such that*

$$\bigcap_{i=1}^{k} \text{conv}(S_i) \neq \emptyset.$$

If no such number exists, let $r_k = \infty$.

We will call $r = r_2$ simply the (not generalized) Radon number. Calder [2] and Eckhoff [3] conjectured the following relationship between the Radon number and the generalized Radon numbers.

[1]The author was supported by the Rényi Doctoral Fellowship and by the NKFIH grant FK132060.

Conjecture 3 (Eckhoff's Partition Conjecture) *For every convexity space, we have*

$$r_k \leq (r-1)(k-1)+1.$$

It was confirmed by Jamison [4] that the conjecture holds if $r_2 = 3$. In general, the best upper bound of r_k in terms of k and $r = r_2$ is $r_k \leq (2k)^{\log_2 r}$ by Jamison [4] and $r_k \leq kr^{r^{r^{\log_2 r}}}$ by Pálvölgyi [5]. However, after being open for more than thirty years, the conjecture itself was refuted by Bukh [1], as he constructed convexity spaces with $r = 4$ and $r_k \geq 3k-1$. As the main motivation for the present work, we will describe his approach in the next section.

The paper is organized as follows. In Section 2 we describe Bukh's approach as a motivation for our work and explain how our work is related to it. Apart from the motivation the rest of the paper is self-contained. In Section 3 we define the Radon number for graph families and give some general results. In Section 4 we show that graph families with Radon number at most three are trivial and in Section 5, as our main result, we characterize graph families with Radon number four. In the end, in Section 6, we describe possible future directions and pose some open questions.

2 Bukh's construction

The results described in this section are from Bukh's preprint [1]. First, we need some notation. Intersecting convex hulls are described using nerves.

Definition 4 *If $(X,\mathcal{C})$ is a convexity space, the B-nerve of a subset $S \subseteq X$ is defined as follows.*

$$\underline{\mathcal{B}}_{\mathcal{C}}(S) = \{\mathcal{A} \subseteq 2^S : \bigcap_{A\in\mathcal{A}} \operatorname{conv}(A) \neq \varnothing\}.$$

In words, $\underline{\mathcal{B}}_{\mathcal{C}}(S)$ consists of all the collections of subsets of S for which collections the convex hulls of the subsets intersect. One can easily describe Radon numbers of convexity spaces using B-nerves.

Claim 5 (Bukh [1], Proposition 6) *The kth generalized Radon number of a convexity space $(X,\mathcal{C})$ is r_k if and only if for every $S \subseteq X$ with $|S| \geq r_k$ there exists a collection in $\underline{\mathcal{B}}_{\mathcal{C}}(S)$ containing k disjoint subsets of S.*

To show that $r_k > t$ for a convexity space it is enough to find an $S \subseteq X$ of size t with suitable $\underline{\mathcal{B}}_{\mathcal{C}}(S)$. The idea is that one can first construct a B-nerve of a t-element set that shows $r_k > t$ and then find a suitable underlying convexity space. But to be able to do that, the B-nerve has to satisfy certain properties.

Lemma 6 (Bukh [1], Lemma 7) *] Given a finite S and $\underline{\mathcal{B}} \subseteq 2^{2^S}$ a family of collections of subsets of S, there exists a convexity space $(X,\mathcal{C})$ with $X \supseteq S$ and $\underline{\mathcal{B}} = \underline{\mathcal{B}}_{\mathcal{C}}(S)$ if and only if $\underline{\mathcal{B}}$ satisfies the following three properties.*

1. *For all $s \in S$ we have $\{A \subseteq S : s \in A\} \in \underline{\mathcal{B}}$.*
2. *If $\mathcal{A}' \subset \mathcal{A} \in \underline{\mathcal{B}}$, then $\mathcal{A}' \in \underline{\mathcal{B}}$.*
3. *If $\mathcal{A} \in \underline{\mathcal{B}}$, then $\{A' \subseteq S : \exists A \in \mathcal{A}$ with $A \subset A'\} \in \underline{\mathcal{B}}$.*

In the proof of Lemma 6, Bukh constructs a concrete convexity space $(X,\mathcal{C})$; we will call it the B-extension of S with respect to $\underline{\mathcal{B}}$. To use this extension to refute Eckhoff's Conjecture, one has to be able to control the Radon number of the B-extension.

Claim 7 (Bukh [1], Proposition 6) *Given a finite S and $\underline{\mathcal{B}} \subseteq 2^{2^S}$, the B-extension of S with respect to $\underline{\mathcal{B}}$ has $r \leq t$ if and only if for all $\mathcal{A}_1,\ldots,\mathcal{A}_t \in \underline{\mathcal{B}}$ there exists a partition $I \cup J = [t]$ with*

$$\left(\bigcap_{i\in I}\mathcal{A}_i\right)\bigcup\left(\bigcap_{j\in J}\mathcal{A}_j\right) \in \underline{\mathcal{B}}.$$

Bukh's counterexample to Eckhoff's Partition Conjecture constructs for each k, a set S with $|S| = 3(k-1)+1$ and a $\underline{\mathcal{B}}$ which (a) satisfies the three properties described in Lemma 6, (b) has no family containing k disjoint subsets and (c) whose B-extension has the property described in Claim 7 with $t = 4$. One of the advantages of his approach is the asymmetry of point (b) which proves $r_k > 3(k-1)+1$ and point (c) which proves $r \leq 4$ using the property described in Claim 7. The main difficulty however lies in the same place. It arises when one wants to check the property described in Claim 7.

In the rest of the paper, our aim is to investigate this property in a rather simplified setting. We hope that after understanding simpler cases, one will be able to construct convexity spaces with $r = 4$ and even larger values of r_k. The main simplification is that instead of a family of collections of arbitrary size subsets of S, we consider only families of collections of subsets of size two in the nerve. In this case, every collection of subsets is an edge set of a graph. Another simplification is more closely related to the particular example Bukh gave in his paper. There every permutation of S was an automorphism of $\underline{\mathcal{B}}$. As this saves a lot of trouble when doing case analysis, we only consider families of graphs that are closed under isomorphisms. Furthermore, we want our graph families to be closed under taking subgraphs in accordance with the second property in Lemma 6, but we drop the requirement of the first property of Lemma 6. The first property is just the requirement that every star is a member of the family, so leaving it out makes our approach a bit more general and probably also a bit more natural. As our last simplification, we do not consider the size of the base set (S) as a parameter to be optimized. We leave it as a further question to consider because it makes the current work more readable.

3 Radon number of graph families

In the rest of the paper, we will consider graph families which are closed under any isomorphism of any graphs in the family and also closed under taking subgraphs. Since the considered graph families are closed under isomorphisms of individual graphs, we have to fix a vertex set as the domain of these isomorphisms. In this paper, we assume that the underlying vertex set is finite but large enough. For our purposes, a vertex set consisting of the first 21 positive integers will be sufficient, but most of our results hold even if the size of the underlying vertex set is just 6. We only consider simple graphs and identify graphs with their edge sets. We will denote the number of edges in a graph G by $|G|$. As the vertex sets of the graphs are labeled, the intersection of any two graphs can be defined as the set of their common edges. The following is our main notion, an analog of the property of B-nerves in Lemma 6.

Definition 8 *We say that a family $\mathcal{G}$ of graphs has Radon number at most r, if no matter how we choose $G_1, \ldots, G_r \in \mathcal{G}$, there is a partition $I \cup J = [r]$ such that*

$$\left(\bigcap_{i\in I} G_i\right) \bigcup \left(\bigcap_{j\in J} G_j\right) \in \mathcal{G}.$$

We will call $\left(\bigcap_{i\in I} G_i\right) \bigcup \left(\bigcap_{j\in J} G_j\right)$ a Radon-major of $G_1, \ldots, G_r$. The smallest number $r(\mathcal{G})$ such that $\mathcal{G}$ has Radon number at most r is the Radon number of the family $\mathcal{G}$.

If we were to follow the problems arising in Bukh's construction more closely, we would consider graph families $\mathcal{G}$ where $r(\mathcal{G}) \leq 4$, every star is a member of the family, but there is no graph with three disjoint edges in the family. We consider instead arbitrary families with bounded Radon numbers to make our approach more general. But first, we determine the Radon number of the family of stars as an example. A star is a graph consisting of any number of edges sharing a common endpoint.

Example 9 *If $\mathcal{G}$ consists of all the stars, then $r(\mathcal{G}) = 4$.*

Proof: We will show that $r(\mathcal{G}) > 3$, then that $r(\mathcal{G}) \leq 4$.

For the proof of $r(\mathcal{G}) > 3$ consider the following three stars: $G_1 = \{\{1,2\},\{1,3\}\}$, $G_2 = \{\{1,2\},\{2,3\}\}$ and $G_3 = \{\{1,3\},\{2,3\}\}$. By symmetry there is essentially one type of partition of $\{G_1, G_2, G_3\}$, for

example $G_1 \cup (G_2 \cap G_3)$, which is a triangle. Since a triangle is not a subgraph of a star, the graphs G_1, G_2, G_3 show that $r(\mathcal{G}) > 3$.

To show that $r(\mathcal{G}) \leq 4$, we need to consider four arbitrary stars $G_1, \ldots, G_4$. If there are two stars, say G_1 and G_2, with the same center, then $(G_1 \cap G_3) \cup (G_2 \cap G_4)$ is also a star with the same center. Otherwise, the intersection of any three stars is empty, and thus $(G_1 \cap G_2 \cap G_3) \cup G_4 = G_4$ provides a good partition. □

Example 10 *If $\mathcal{G}$ consists of all the n-cliques and their subgraphs, then $r(\mathcal{G}) = n + 1$.*

Proof: First, we show that $r(\mathcal{G}) \leq n + 1$. Let $G_1, \ldots, G_n$ be cliques on n vertices and consider the sequence $\cap_{i=1}^{j} V(G_i)$ for $1 \leq j \leq n + 1$, where $V(G_i)$ is the vertex set of G_i. If there exists a j with $\cap_{i=1}^{j} V(G_i) = \cap_{i=1}^{j+1} V(G_i)$, then $\cap_{i=1}^{j} G_i \subset G_{j+1}$ and thus $G = (\cap_{i=1}^{j} G_i) \cup (\cap_{i=j+1}^{n+1} G_i) \subset G_{j+1}$, which implies $G \in \mathcal{G}$. Otherwise we have $|\cap_{i=1}^{n} V(G_i)| \leq 1$, and thus $(\cap_{i=1}^{n} G_i) \cup G_{n+1} = G_{n+1} \in \mathcal{G}$.

Now we show that $r(\mathcal{G}) > n$. Let $G_1, \ldots, G_n$ be cliques on n vertices where the vertex set of G_i is $[n+1] \setminus \{i\}$. No matter how we partition $[n+1]$ into two nonempty subsets $I \uplus J = [n+1]$, the graph $G = (\cap_{i \in I} G_i) \cup (\cap_{j \in J} G_j)$ has $n+1$ vertices, but none of them is isolated. Thus, G is not a subgraph of an n-clique. □

Our main tool in considering Radon numbers of graph families will be an analog of Helly's Theorem. We need two preliminary definitions to state it.

Definition 11 *The h-Helly closure $\mathcal{H}_h(\mathcal{G})$ of a graph family $\mathcal{G}$ consists of all graphs H for which every at most h–edge subgraph of H is an element of $\mathcal{G}$.*

As an example, if $\mathcal{G}$ consists of all the stars with at most h edges, then its h-Helly closure consists of all the stars with an arbitrary number of edges.

Definition 12 *We say that a graph family $\mathcal{G}$ implies a graph G under the condition $r(\mathcal{G}) \leq t$ if every graph family $\mathcal{G}' \supseteq \mathcal{G}$ with $r(\mathcal{G}') \leq t$ contains G as well. In notation $\mathcal{G} \overset{r \leq t}{\Longrightarrow} G$. We also write $\mathcal{G} \overset{r \leq t}{\Longrightarrow} \mathcal{F}$ for a graph family $\mathcal{F}$, if $\mathcal{G} \overset{r \leq t}{\Longrightarrow} F$ for every $F \in \mathcal{F}$.*

We will use that this kind of implication is a transitive relation if t is fixed. The following is an analog of Helly's Theorem for graph families.

Lemma 13 (Helly Property) *For every graph family $\mathcal{G}$ and every integer $h > 0$ we have*

$$\mathcal{G} \overset{r \leq h+1}{\Longrightarrow} \mathcal{H}_h(\mathcal{G}).$$

Proof: The proof of Lemma 13 mimics Radon's proof [6] of Helly's Theorem. Let $\mathcal{G}'$ be a family containing $\mathcal{G}$ and satisfying $r(\mathcal{G}') \leq h + 1$. We will prove $H \in \mathcal{G}'$ for every $H \in \mathcal{H}_h(\mathcal{G})$ by induction on the number of edges in H. If H has h elements, then it is equal to some $G \in \mathcal{G}$ and hence a member of $\mathcal{G}'$. For the induction step, assume, that H has more than h edges and that every proper subgraph of H is a member of $\mathcal{G}'$. Let $H_1, \ldots, H_{h+1}$ be different subgraphs of H, each containing exactly $|H| - 1$ edges. No matter how we partition $[h+1]$ into two nonempty subsets $I \uplus J = [h+1]$, we have

$$\left(\bigcap_{i \in I} H_i\right) \bigcup \left(\bigcap_{j \in J} H_j\right) = H,$$

showing that $\mathcal{G}'$ must contain H as well. □

Definition 14 *A graph family $\mathcal{G}$ is t-Radon closed, if $r(\mathcal{H}_{t-1}(\mathcal{G})) = t$.*

Claim 15 *If $r(\mathcal{G}) \leq t$ for a graph family $\mathcal{G}$, and a graph family $\mathcal{G}'$ consists of graphs with at most $t-2$ edges, then $r(\mathcal{G} \cup \mathcal{G}') \leq t$ as well.*

PROOF: Let $G_1, \ldots, G_t \in \mathcal{G} \cup \mathcal{G}'$. If there is any G_i with at most $t-2$ edges, then there are two different $I, I' \subset [t]$ with $|I| = |I'| = t-1$, $i \in I \cap I'$ and $\cap_{j \in I} G_j = \cap_{j \in I'} G_j$ by the pigeonhole principle. In particular, $\cap_{j \in I} G_j \subset G_k$ if $\{k\} = [t] \setminus I$. I and $J = \{k\}$ provides a good partition, because $\cap_{j \in I} G_j \cup G_k \subseteq G_k$.

If there is no G_i among $G_1, \ldots, G_t$ with at most $t-2$ edges, then $G_1, \ldots, G_t \in \mathcal{G}$ and there exists a good partition $I \uplus J = [t]$ with $(\cap_{i \in I} G_i) \cup (\cap_{j \in J} G_j) \subset G \in \mathcal{G}$. □

Corollary 16 *Every graph family $\mathcal{G}$ with $r(\mathcal{G}) = t$ is a union of a family of graphs with at most $(t-2)$ edges and a $(t-1)$-Helly closure of a family of graphs with exactly $t-1$ edges.*

We characterize graph families with Radon number at most three as a warm-up.

4 Graph families with Radon number at most 3 are trivial

Claim 17 *If $r(\mathcal{G}) = 2$ for a graph family, then either $\mathcal{G}$ is the empty family or $\mathcal{G}$ contains the empty graph with no edges or $\mathcal{G}$ contains all the graphs.*

PROOF: If $r(\mathcal{G}) = 2$, then the Radon-major of any G_1, G_2 is $G_1 \cup G_2$, thus $\mathcal{G}$ is closed under taking the union of two graphs in $\mathcal{G}$. Since $\mathcal{G}$ is also closed under taking any isomorphic copy of any of its graphs, if it contains a graph with at least one edge, then it contains all the graphs. □

Proposition 18 *If $r(\mathcal{G}) = 3$ for a graph family $\mathcal{G}$, then $\mathcal{G}$ consists of all the graphs with exactly one edge.*

Note that there are only two different non-isomorphic simple graphs with exactly two edges, a path (denoted from now by the symbol $\wedge$) and a graph consisting of two disjoint edges (denoted by $=$). We show that any one of them contains the other in its 2-Helly closure. The proof of Proposition 18 will be an easy corollary.

Claim 19 $\{\wedge\} \overset{r \leq 3}{\Longrightarrow} =$.

PROOF: First, note that every two-edge subgraph of a triangle is isomorphic to $\wedge$. By Lemma 13 it follows that $\{\wedge\} \overset{r \leq 3}{\Longrightarrow} K_3$. We will further show, that $K_3 \overset{r \leq 3}{\Longrightarrow} =$. Let G_i be the triangle with vertices $\{i\} \setminus \{1,2,3,4\}$ and consider G_1, G_2, G_3. By symmetry, there is only one type of partition, whose Radon-major is isomorphic to $G_1 \cup (G_2 \cap G_3)$. But it contains two disjoint edges, namely $\{1,4\}$ from $G_2 \cap G_3$ and $\{2,3\}$ from G_1. Since every Radon-major of G_1, G_2, G_3 contains a subgraph isomorphic $=$, we can conclude, that $\{\wedge\} \overset{r \leq 3}{\Longrightarrow} \{K_3\} \overset{r \leq 3}{\Longrightarrow} =$, finishing the proof of the Claim. □

Claim 20 $\{=\} \overset{r \leq 3}{\Longrightarrow} \wedge$.

PROOF: By Lemma 13 we have $\{=\} \overset{r \leq 3}{\Longrightarrow} \equiv$, where $\equiv$ is the graph with three disjoint edges. Let $G_1 = \{12, 45, 78\}$, $G_2 = \{12, 56, 89\}$ and $G_3 = \{23, 45, 89\}$ (we denote edges of the form $\{i, j\}$ simply by ij). No matter how we partition $\{G_1, G_2, G_3\}$ into two nonempty subsets, $\wedge$ will be a subgraph of any Radon-major. □

PROOF:[of Propositoin 18] The family $\mathcal{G}$ which contains all the graphs with exactly one edge has $r(\mathcal{G}) > 2$ by Claim 17 and has $r(\mathcal{G}) \leq 3$ by Claim 15.

On the other hand, the union of Claims 19 and 20 shows that if a graph family contains any graph with at least two edges, then its 2-Helly closure contains all the graphs. By Lemma 13, $\mathcal{G}$ must contain all the graphs if we have $r(\mathcal{G}) \leq 3$, but in this case $r(\mathcal{G}) = 2$, finishing the proof of Proposition 18. □

In the following pages we will use the scheme of the proof of Claim 19 many times. We started with some small graphs and find bigger ones in their h-Helly closure. Then we take $h+1$ of the bigger ones and show that there is no Radon major of them without a specific subgraph. If possible, we want the configuration of the four graphs to be very symmetric to avoid dealing with many cases.

5 Graph families with Radon number 4

We will use the following symbols for the five isomorphism classes of graphs with exactly three edges: $\equiv, \curlywedge, \triangle, \eqsim, \sqcap$. The first denotes a graph with three disjoint edges, the second a star with three edges, the third a triangle, the fourth a disjoint union of a path with two edges and an edge, and the fifth a path with three edges.

Theorem 21 *If a graph family $\mathcal{G}$ with Radon number at most 4 contains any of the four families*

$$\{\equiv\}, \{\curlywedge, \eqsim\}, \{\curlywedge, \triangle, \sqcap\}, \{\triangle, \eqsim, \sqcap\},$$

then it contains all the graphs. If a graph family contains neither of the above listed families, then its 3-Helly closure has Radon number 4, or equivalently, the family is 4-Radon closed.

We will prove the first part of Theorem 21 in Subsection 5.1 and the second part in Subsection 5.2.

5.1 Implications

The proof of the first part of Theorem 21 is broken down into seven small claims.

Claim 22 $\{\equiv\} \overset{r\leq 4}{\Longrightarrow} \eqsim$.

Proof: There are seven different 2-partitions of $\{1,2,3,4\}$, namely 1|234, 2|134, 3|124, 4|123, 12|34, 13|24 and 14|23. Let $I_1, J_1, \ldots, I_7, J_7$ be the subsets of these partitions in the previous order. For example $I_1 = \{1\}$, $J_1 = \{2,3,4\}$, $I_5 = \{1,2\}$ or $J_7 = \{2,3\}$. Note that a graph consisting of any number of disjoint edges is part of the 3-Helly closure of $\{\equiv\}$. Now consider seven disjoint copies of $\wedge$, each corresponding to a different 2-partition of $\{1,2,3,4\}$. Name the two edges of ks copy as $\{e_i^{(k)}, e_j^{(k)}\}$ and let $G_1, \ldots, G_4$ be such that $G_\ell = \{e_i^{(k)} : \ell \in I_k\} \cup \{e_j^{(k)} : \ell \in J_k\}$. No matter what partition $I \uplus J = \{1,2,3,4\}$ we take, the corresponding copy of $\wedge$ will be in $\left(\bigcap_{i\in I} G_i\right) \bigcup \left(\bigcap_{j\in J} G_j\right)$. The last observation is that at least one of the intersections will contain an edge from a different copy of $\wedge$, resulting in a copy of $\eqsim$ in every Radon-major of $G_1, \ldots, G_4$. □

Claim 23 $\{\equiv\} \overset{r\leq 4}{\Longrightarrow} \triangle, \curlywedge, \sqcap$.

Proof: Let F be one of the three graphs $\triangle, \curlywedge, \sqcap$. Consider seven disjoint copies of F and denote the edges of the kth one by $e_1^{(k)}, e_2^{(k)}, e_3^{(k)}$. Now let G_1, G_2, G_3 and G_4 be defined as follows. Let $I_1, \ldots, J_7$ subsets of different partitions of $\{1,2,3,4\}$ as in the proof of Claim 22, and let $G_\ell = \{e_1^{(k)} : \ell \in I_k\} \cup \{e_2^{(k)} : \ell \in I_k\} \cup \{e_3^{(k)} : \ell \in J_k\}$. This way every G_ℓ consists of disjoint edges and 2-edge paths and thus $\{\equiv\} \overset{r\leq 4}{\Longrightarrow} G_\ell$. The proof ends with the observation, that no matter how we partition $\{1,2,3,4\}$ into $I \uplus J$, one of the seven copies of F will be a subgraph of $\left(\bigcap_{i\in I} G_i\right) \bigcup \left(\bigcap_{j\in J} G_j\right)$. □

Claim 24 $\{\curlywedge, \triangle, \sqcap, \eqsim\} \overset{r\leq 4}{\Longrightarrow} \equiv$.

Proof: First observe, that $\{\curlywedge, \triangle, \sqcap, \eqsim\} \overset{r\leq 4}{\Longrightarrow} K_5$ by the Helly Property (Lemma 13). Now let G_i be the five-clique on $\{1, \ldots, 6\}$ avoiding vertex i. By symmetry, checking partitions 12|34 and 1|234 are sufficient. In the first case, the edges $12, 34, 56$ are present in $(G_1 \cap G_2) \cup (G_3 \cap G_4)$. In the second case, edges $16, 23, 45$ are present in $G_1 \cup (G_2 \cap G_3 \cap G_4)$. Thus, $\{G_1, G_2, G_3, G_4\}$ has no Radon-major without $\equiv$ as a subgraph. □

Claim 25 $\{\curlywedge, \triangle, \sqcap\} \overset{r\leq 4}{\Longrightarrow} \eqsim$.

Proof: Now $\{\curlywedge, \triangle, \sqcap\} \overset{r\leq 4}{\Longrightarrow} K_4$ by the Helly Property. Let G_i be a four-clique on $\{1, \ldots, 5\}$ avoiding vertex i. By symmetry, checking partitions 12|34 and 1|234 are sufficient. In the first case $\{34, 45, 12\} \subseteq (G_1 \cap G_2) \cup (G_3 \cap G_4)$, in the second case $\{23, 34, 15\} \subseteq G_1 \cup (G_2 \cap G_3 \cap G_4)$, showing that there is no Radon-major of $\{G_1, G_2, G_3, G_4\}$ without $\eqsim$ as a subgraph. □

Claim 26 $\{\curlywedge, \eqsim, \sqcap\} \overset{r\leq 4}{\Longrightarrow} \triangle$.

Proof: Observe that $\{\curlywedge, \eqsim, \sqcap\} \overset{r\leq 4}{\Longrightarrow} K_{2,3}$ by the Helly Property. Let $G_1, \ldots, G_4$ be $K_{2,3}$s on $\{1, \ldots, 5\}$ with G_i being the complete bipartite graph between $\{i, 5\}$ and $\{i, 5\} \setminus \{1, \ldots, 5\}$. As before, due to symmetry, we only need to check partitions 12|34 and 1|234. In the first case $\{12, 15, 25\} \subseteq (G_1 \cap G_2) \cup (G_3 \cap G_4)$ and in the second case $\{12, 15, 25\} \subseteq G_1 \cup (G_2 \cap G_3 \cap G_4)$. □

Claim 27 $\{\curlywedge, \eqsim\} \overset{r\leq 4}{\Longrightarrow} \sqcap$.

Proof: By Lemma 13 we have $\{\curlywedge, \eqsim\} \overset{r\leq 4}{\Longrightarrow} S_l \uplus S_k$, any union of two vertex-disjoint stars. We will consider four such graph on vertices $\{0, 1, \ldots, 8\}$, all with centers 0 and 8. Define $I_1, J_1, \ldots, I_7, J_7$ as in the proof of Claim 22 and let $G_\ell = \{\{0, k\} : \ell \in I_k\} \cup \{\{k, 8\} : \ell \in J_k\}$ for $\ell \in \{1, 2, 3, 4\}$. Now if we take the kth partition $I_k \uplus J_k = \{1, 2, 3, 4\}$, the graph $\left(\bigcap_{i\in I_k} G_i\right) \bigcup \left(\bigcap_{j\in J_k} G_j\right)$ will contain the edges $\{0, k\}, \{k, 8\}$ and at least one more edge, resulting in a copy of $\sqcap$ in every Radon-major. □

Claim 28 $\{\triangle, \eqsim, \sqcap\} \overset{r\leq 4}{\Longrightarrow} \curlywedge$.

Proof: Observe that any cycle with five edges, any graph consisting of two disjoint triangles, and any path with four edges are part of the 3-Helly closure of $\{\triangle, \eqsim, \sqcap\}$. Now consider the following graphs: G_1 a five cycle with vertices $1, 2, 3, 4, 5$, G_2 a disjoint union of two triangles with vertices $1, 2, 3$ and $4, 5, 6$, G_3 a path with five vertices $2, 3, 4, 6, 5$ and G_4 another path with five vertices $1, 3, 4, 5, 6$. As there is no symmetry in the configuration, we have to check all seven possible partitions of them. We will show undesirable 3-edge subgraphs in the Radon major in every case. We have $\{15, 45, 56\} \subset G_1 \cup (G_2 \cap G_3 \cap G_4)$ in the first case, $\{13, 23, 34\} \subset G_2 \cup (G_1 \cap G_3 \cap G_4)$ in the second case, $\{34, 45, 46\} \subset G_3 \cup (G_1 \cap G_2 \cap G_4)$ in the third, $\{13, 23, 34\} \subset G_4 \cup (G_1 \cap G_2 \cap G_3)$ in the fourth, $\{13, 23, 34\} \subset (G_1 \cap G_3) \cup (G_2 \cap G_4)$ in the fifth and $\{34, 45, 46\} \subset (G_1 \cap G_4) \cup (G_2 \cap G_3)$ in the sixth case. In the last case $\{12, 34, 56\} \subset (G_1 \cap G_2) \cup (G_3 \cap G_4)$ which is a copy of $\equiv$, but we have seen, that $\{\equiv\} \overset{r\leq 4}{\Longrightarrow} \curlywedge$. □

Now we are ready to prove the first part of Theorem 21, as it is an easy consequence of the previous seven claims and Lemma 13.

Proof:[of the first part of Theorem 21] If $\{\equiv\} \subseteq \mathcal{G}$, then all the other graphs with exactly three edges are elements of $\mathcal{G}$ by Claims 22 and 23. From here Lemma 13 implies that every graph is part of $\mathcal{G}$.

If $\{\curlywedge, \eqsim\} \subseteq \mathcal{G}$, then we can combine Claims 24, 26 and 27 and the previous paragraph to reach the same conclusion. For the cases where $\{\curlywedge, \triangle, \sqcap\}$ or $\{\triangle, \eqsim, \sqcap\}$ is a subfamily of $\mathcal{G}$, we can combine Claim 25 or Claim 28 with Claim 24 and the previous paragraph. □

5.2 Radon-closed families

Let $\mathcal{G}$ be an arbitrary graph family. To show that $\mathcal{G}$ has Radon number at most 4, we have to exclude the possibility of the existence of $G_1, \ldots, G_4 \in \mathcal{G}$ without a Radon-major in $\mathcal{G}$. The following claims show some necessary properties of such quadruples of graphs.

Claim 29 *If $G_1, \ldots, G_4 \in \mathcal{G}$ has no Radon-major in $\mathcal{G}$, then the 3-wise intersections of them must be nonempty and distinct.*

Proof: If a three-wise intersection is empty, for example $G_1 \cap G_2 \cap G_3 = \varnothing$, then G_4 is a Radon major, as $(G_1 \cap G_2 \cap G_3) \cup G_4 = G_4$.

If two of the three-wise intersections are equal, for example $G_1 \cap G_2 \cap G_3 = G_2 \cap G_3 \cap G_4$, then G_4 is a Radon-major, as $(G_1 \cap G_2 \cap G_3) \cup G_4 = (G_2 \cap G_3 \cap G_4) \cup G_4 = G_4$. □

Claim 30 *If $G_1, \ldots, G_4$ are arbitrary graphs, then*

$$\bigcup_{i=1}^{4} \bigcap_{j \neq i} G_j \in \mathcal{H}_3(\{G_1, \ldots, G_4\}).$$

Proof: If we choose 3 edges from the union of the 3-wise intersections, there will be a G_i among $G_1, \ldots, G_4$, which contains all the 3 edges. □

Given four graphs $G_1, \ldots, G_4$ and subset $I \subset \{1,2,3,4\}$, we will say that an edge $e \in \cup_{i=1}^{n} G_i$ is I-type, if $e \in \bigcap_{i \in I} G_i$ but for every $J \supsetneq I$ we have $e \notin \bigcap_{j \in J} G_j$.

In this sense Claim 29 states that given $G_1, \ldots, G_4 \in \mathcal{G}$ with no Radon-major there must be at least one I-type edge for every $I \subset [4]$, $|I| = 3$.

Claim 31 *If $G_1, \ldots, G_4 \in \mathcal{H} = \mathcal{H}_3(\mathcal{G})$ has no Radon-major in $\mathcal{H}$, then there is either a $\{1,2\}$-type edge or a $\{3,4\}$-type edge in $\cup_i G_i$.*

Proof: If there is no $\{1,2\}$-type edge, then $G_1 \cap G_2 = (G_1 \cap G_2 \cap G_3) \cup (G_1 \cap G_2 \cap G_4) \in \mathcal{H}$ by Claim 30. We must have an edge e in $(G_3 \cap G_4) \setminus ((G_1 \cap G_3 \cap G_4) \cup (G_2 \cap G_3 \cap G_4))$, otherwise $(G_1 \cap G_2) \cup (G_3 \cap G_4) \in \mathcal{H}$ would be a Radon-major. □

Claim 32 *If all the graphs in $\mathcal{H} = \mathcal{H}_3(\mathcal{G})$ have at most 4 edges, then $r(\mathcal{H}) \leq 4$.*

Proof: Let $G_1, \ldots, G_4 \in \mathcal{H}$ and consider first the 3-wise intersections of them. If $G_1, \ldots, G_4$ have no Radon-major in $\mathcal{H}$, then by Claim 29 every G_i must contain at least 3 edges, one in each distinct 3-wise intersection where G_i is one of the intersecting graphs.

Now consider the 2-wise intersections. By Claim 31 at least three of them must be of size at least three. It follows from the pigeon-hole principle that at least one of the G_is have at least five edges. □

Corollary 33 *The 3-Helly closures of the families $\{\sqcap\}, \{\doteq\}, \{\triangle\}$ and $\{\sqcap, \triangle\}$ have Radon number 4.*

Proof: In all the mentioned families the 3-Helly closure contains only graphs with at most 4 edges. We will show that by showing that the 3-edge graphs in the families can not be extended with more than one edge while staying in the 3-Helly closure of the family. This is sufficient since every graph with at least three edges in the 3-Helly closure contains at least one of the listed 3-edge graphs as a subgraph.

In the case of $\{\sqcap\}$, there is only one way to ad an edge to $\sqcap$ without creating other types of 3-edge subgraphs and this creates a C_4, which is a maximal graph in the 3-Helly closure.

In the case of $\{\eqslantless\}$, there is also only one way to extend $\eqslantless$ with an edge and this creates a graph consisting of two disjoint paths each with two edges. This graph can not be extended without leaving the 3-Helly closure of $\{\eqslantless\}$.

In the case of $\{\triangle\}$, the triangle itself is a maximal graph in the Helly-closure which can not be extended.

In the case of $\{\sqcap, \triangle\}$, we either start with a $\sqcap$ or with a $\triangle$, but even if the other 3-edge graph is allowed, there is no other way to extend these graphs than to extend $\sqcap$ into a C_4. □

Corollary 34 *The 3-Helly closures of the families* $\{\curlywedge\}, \{\curlywedge, \sqcap\}$ *and* $\{\curlywedge, \triangle\}$ *has Radon number* 4.

Proof: Let $\mathcal{G}$ be any of the families $\{\curlywedge\}, \{\curlywedge, \sqcap\}$ or $\{\curlywedge, \triangle\}$. Suppose for contradiction, that $G_1, \ldots, G_4 \in \mathcal{H} = \mathcal{H}_3(\mathcal{G})$ has no Radon-major in $\mathcal{H}$ and look at $G = \bigcup_{i=1}^{4} \bigcap_{j \neq i} G_j$. It is a member of $\mathcal{H}$ by Claim 30 and we have $|G| \geq 4$ and $|G \cap G_i| \geq 3$ for all i by Claim 29.

In the case of $\mathcal{G} = \{\curlywedge\}$, G is a star with at least four edges. $G_1, \ldots, G_4$ are also stars and moreover, they have the same center, since $|G \cap G_i| \geq 3$ for all i. Now even their union is in $\mathcal{H}$, contradicting the assumption that they have no Radon-major in $\mathcal{H}$.

If $\mathcal{G} = \{\curlywedge, \sqcap\}$, then G is either a star or a C_4. The first case brings us back to the previous paragraph. If $G = C_4$, then all the G_is are subgraphs of G with at least 3 edges and their union is exactly G.

If $\mathcal{G} = \{\curlywedge, \triangle\}$, then G must be a star with at least four edges and the rest of the proof goes as in the case when $\mathcal{G} = \{\curlywedge\}$. □

Corollary 35 *The 3-Helly closures of the families* $\{\eqslantless, \triangle\}$ *and* $\{\eqslantless, \sqcap\}$ *has Radon number* 4.

Proof: If $\mathcal{G} = \{\eqslantless, \triangle\}$, then all the maximal graphs in $\mathcal{H} = \mathcal{H}_3(\mathcal{G})$ are graphs consisting of two disjoint triangles. Suppose for contradiction that there exists $G_1, \ldots, G_4 \in \mathcal{H}$ without a Radon-major in $\mathcal{H}$. We might assume that all of them are maximal graphs in $\mathcal{H}$. Let $G_i = T_i \cup T_i'$ a disjoint union of two triangles. The graphs $G_1, \ldots, G_4$ can not all have a common triangle, since in this case they can not have distinct 3-wise intersections, which contradicts Claim 29. If they do not have a common triangle, there are two of them without a common triangle. Let them be G_3 and G_4. By Claim 31 G_1 and G_2 intersect in at least 3 edges, so they have a common triangle, denote it by T. By Claim 29 there must be edges $e \in (G_2 \cap G_3 \cap G_4) \setminus T$ and $e' \in (G_1 \cap G_3 \cap G_4) \setminus T$. The edges e and e' are disjoint, because G_3 and G_4 don't have a common triangle. We may assume without loss of generality, that $e \in T_3 \cap T_4$ and $e' \in T_3' \cap T_4'$. But the edges e and e' are vertex-disjoint from T, so $G_1 \cap G_2 \cap G_3$ and $G_1 \cap G_2 \cap G_4$ are empty, contradicting Claim 29.

The case $\mathcal{G} = \{\eqslantless, \sqcap\}$ is even more complicated, because there are three types of maximal graphs in $\mathcal{H} = \mathcal{H}_3(\mathcal{G})$. These maximal graphs are a cycle with four edges (C_4), a cycle with five edges (C_5) and a disjoint union of two paths each with two edges ($2P_3$). Suppose for contradiction that there are $G_1 \ldots, G_4 \in \mathcal{H}$ without a Radon-major in $\mathcal{H}$. We may assume that each of them is isomorphic to one of $C_4, C_5, 2P_3$ and also that $G = \bigcup_{i=1}^{4} \bigcap_{j \neq i} G_j$ isomorphic to an at least 4-edge subgraph of one of C_4, C_5 and $2P_3$ by Claim 30. Observe that all the intersections $G_i \cap G$ must be distinct and must have size at least 3 by Claim 29. In all three cases when G is isomorphic to C_4, C_5, or $2P_3$ the previous facts will contradict Claim 31. □

Proof:[of the second part of Theorem 21]

If $\mathcal{G}$ does not contain any graph with at least three edges, then $r(\mathcal{G}) \leq 4$ by Claim 15.

If $\mathcal{G}$ does contain graphs with at least three edges, $r(\mathcal{G}) = 4$ and $\mathcal{G}$ does not have $\{\equiv\}, \{\curlywedge, \eqslantless\}, \{\curlywedge, \triangle, \sqcap\}$ or$\{\triangle, \eqslantless, \sqcap\}$ as subfamilies, then its 3-Helly closure equals to the 3-Helly closure of one of the remaining nonempty subfamilies of the five isomorphism classes of graphs with three edges. These subfamilies are $\{\sqcap\}, \{\eqslantless\}, \{\triangle\}, \{\curlywedge\}, \{\sqcap, \triangle\}, \{\curlywedge, \sqcap\}, \{\curlywedge, \triangle\}, \{\eqslantless, \triangle\}$ and $\{\eqslantless, \sqcap\}$. We have seen in Corollaries 33, 34 and 35 that for all the above-listed families the Radon number r of the 3-Helly closures is 4. □

6 Open questions

There are a lot of questions one can ask about the Radon number of different (graph) families. We chose the following two with the hope that investigating them will bring us closer to improving Bukh's counterexample to Eckhoff's Conjecture. First, the collections in B-nerves are not necessarily edge sets of graphs, but of general set systems. The next step towards this general setting might be to consider 3-uniform hypergraphs instead of graphs with the straightforward generalization of the definition of Radon numbers for graph families.

Problem 36 *Characterize families of* 3*-uniform hypergraphs with Radon number at most* 4*.*

One might also try to find analogs of Bukh's construction with larger Radon numbers. The first step might be to consider graph families with Radon number 5.

Problem 37 *Characterize graph families* $\mathcal{G}$ *with Radon number* $r(\mathcal{G}) = 5$*.*

Acknowledgement

The author would like to thank Dömötör Pálvölgyi for a lot of interesting and encouraging discussions during the research.

References

[1] B. Bukh, Radon partitions of convexity spaces, *arXiv preprint, arXiv:1009.2384* (2010)

[2] J.R. Calder, Some elementary properties of interval convexities, *J. Lond. Math. Soc.* **3,** 422–428 (1971)

[3] J. Eckhoff, Radon's theorem revisited, *Contributions to geometry* 164-185. (Siegen, 1978)

[4] R.E. Jamison-Waldner, Partition numbers for trees and ordered sets, *Pac. J. Math.* **96**(1), 115–140 (1981)

[5] D. Pálvölgyi, Radon numbers grow linearly, *Discrete and Computational Geometry* **68**(1), 165-171. (2022)

[6] J. Radon, Mengen konvexer Körper, die einen gemeinsamen Punkt enthalten, *Math. Ann.* **83**(1-2), 113–115 (1921)

[7] H. Tverberg, A generalization of Radon's theorem, *J. Lond. Math. Soc.* **41,** 123-128 (1966)

[8] M.L.J. van de Vel, Theory of Convex Structures, *Elsevier* (1993)

Rigid planar subgraphs in the triangulations of the double torus

Viktória E. Kaszanitzky[1]

Department of Computer Science and
Information Theory
Budapest University of Technology and
Economics
1111 Budapest, Műegyetem rkp. 3., Hungary
and
ELKH-ELTE Egerváry Research Group on
Combinatorial Optimization
1111 Budapest, Pázmány P. s. 1/C, Hungary
kaszanitzky@cs.bme.hu

Abstract: In [9] Nevo and Tarabykin proved that every triangulation of the torus, the projective plane and the Klein bottle has a spanning planar Laman subgraph. Combining this with [5], a paper of Király, the result implies, that every triangulation of the above mentioned surfaces can be realised as an infinitesimally rigid framework with just 26 vertex locations in the plane.

In this paper we prove the corresponding result for the double torus.

Keywords: combinatorial rigidity, rigid realisations, triangulated surfaces

1 Introduction

1.1 Rigidity in the plane

A d-dimensional *framework* (G,p) is a graph $G=(V,E)$ together with a map $p: V \to \mathbb{R}^n$. The *rigidity matrix* $R(G,p)$ of size $|E| \times d|V|$ has one row corresponding to each of its edges, and d columns corresponding to each of its vertices. The row of $uv \in E$ has entries $p(u)-p(v)$ in the columns of u and $p(v)-p(u)$ in the columns of v. The rest of the entries in this row are zeros. An *infinitesimal motion* of (G,p) is an assignment $m: V \to \mathbb{R}^d$, such that $(p(u)-p(v))(m(u)-m(v))=0$ for every edge $uv \in E$. Equivalently, if $R(G,p)m=0$. An infinitesimal motion m is *trivial* if $m(v)=Sp(v)+t$ holds for all $v \in V$, for some $d \times d$ skew-symmetric matrix S and some vector $t \in \mathbb{R}^d$. (G,p) is *infinitesimally rigid* in $\mathbb{R}^d$ if all of its infinitesimal motions are trivial.

A d-dimensional framework (G,p) is *generic* if the $d|V|$ coordinates of its points are algebraically independent over $\mathbb{Q}$. Graph G is said to be *rigid* in $\mathbb{R}^d$ if every (or equivalently, if some) generic d-dimensional framework (G,p) is infinitesimally rigid in $\mathbb{R}^d$. G is *minimally rigid* in $\mathbb{R}^d$, if G is rigid, but after the deletion of any edge it is no longer rigid. For $d=2$ a theorem of Laman characterises the minimally rigid graphs:

Theorem 1 *[6] Graph $G=(V,E)$ is minimally rigid in $\mathbb{R}^2$ if and only if $|E|=2|V|-3$ and for every subgraph $G'=(V',E')$ with at least two vertices $|E'| \leq 2|V'|-3$.*

[1]Research is supported by then and by the Hungarian National Research, Development and Innovation Office (NKFIH), grant number FK128673.

1.2 Triangulations of surfaces

A triangulation of a surface is a simple graph $G = (V, E)$ that can be embedded into the surface such that the edges do not intersect each other and every face is a triangle. A triangulation may be reduced with an *edge contraction*, where we take an edge uv and delete the vertex u and we add edges vu_i for every u_i for which $uu_i \in E$ and $vu_i \notin E$. If a triangulation does not have an edge which can be contracted such that the resulting graph is also a triangulation of the same surface then it is called *irreducible*. The inverse operation of the edge contraction is the *vertex split*, which takes two edges uv, uw in the embedding. It doubles u, thus creates u_1 and u_2, connects them with an edge, connects both u_1 and u_2 with v and w. The neighbours of u form a cycle which is divided into two paths by v and w. The vertices of one of these paths are connected with u_1 and the vertices on the other path are connected with u_2. From the set of the irreducible triangulations of a surface every triangulation of the same surface can be generated with a series of vertex splits.

The set of irreducible triangulations is determined for a number of surfaces, see [1], [7], [8], [11], [12]. Using these irreducible triangulations Nevo and Tarabykin proved the following:

Theorem 2 *[9] Every triangulation of the projective plane contains a spanning disc. Every triangulation of the torus contains a spanning cylinder. Every triangulation of the Klein bottle contains either a cylinder, or a connected sum of two triangulated discs along a triangle.*

We shall need some properties of surfaces with higher genus for proving similar results for them.

Let $K = v_1v_2 \dots v_k$ be a cycle of a graph G embedded in a surface S and let C be the curve corresponding to the cycle in the embedding. We say that K is *separating*, if $S - C$ is disconnected. The cycle $v_1v_2 \dots v_k$ is *non-contractible*, if none of the two components of $G - K$ is planar. We will call non-contractible separating cycles *NSC* for short.

Ellingham, Zha, and Jennings proved the following:

Theorem 3 *[4] Every triangulation of the double torus has an NSC.*

Sulanke proved the same statement for multiple surfaces:

Theorem 4 *[11] Every triangulation of the double torus, the Klein bottle, the triple cross surface or the quadruple cross surface has an NSC.*

1.3 Triangulations with few vertex locations in the plane

For a graph $G = (V, E)$ rigid in $\mathbb{R}^d$, what is the smallest cardinality of a set $P \subseteq \mathbb{R}^d$, such that an infinitesimally rigid framework (G, p) with $p : V \to P$ exists? Similarly, the same question can be asked for a family $\mathcal{G}$ of graphs rigid in $\mathbb{R}^d$, namely how small can a set $P \subseteq \mathbb{R}^d$ be for which every graph in $\mathcal{G}$ has an infinitesimally rigid realisation where all vertex positions are in P?

It is known, that for $d = 1$ and the family of connected graphs (which are exactly the rigid graphs in one dimension) the answer is 2. For $d \geq 2$ and the family of every rigid graph no such P exists, see [2].

A result of Fogelsanger [3] says that for every g a triangulation of a surface with genus g is rigid in three dimensions (and also in two dimensions). It is a result of Király [5] that every triangulation of a surface with genus g has an infinitesimally rigid framework with $O(\sqrt{g})$ vertex locations. If we take $\mathcal{G}$ as the family of triangulations of surfaces for a fixed genus g we can ask if there is a constant upper bound for the size of the number of different locations.

For the sphere Király proved that a constant upper bound of 26 exists.

Theorem 5 *[5] Let $A \subseteq \mathbb{R}^2$ be a generic set with $|A| = 26$. Then, for every planar graph $G = (V, E)$ which is rigid in $\mathbb{R}^2$, there exists an infinitesimally rigid realization $p : V \to A$ of G.*

Nevo and Tarabykin shoved that the same upper bound works for the torus, the Klein bottle and the projective plane by proving that all of their triangulations contain a spanning planar Laman subgraph and from the work of Király it is known that 26 is an upper bound for planar Laman graphs.

2 Preliminaries

We will need the following simple lemmas in the proof of the main result. The following can be easily proven by building up the graph using planar vertex splits.

Lemma 6 *Let G be a planar graph in which the vertex sets of faces with boundaries longer than three are pairwise disjoint. Then G is rigid.*

The next lemma is a well-known statement, see [10] for a reference.

Lemma 7 *Let G and H be two rigid graphs. Let $v_1, \ldots, v_k \in V(G)$ and $u_1, \ldots, u_k \in V(H)$ vertices and $k \geq 2$. The graph we get by identifying u_i with v_i for every $1 \leq i \leq k$ (thus gluing the two graphs) is rigid.*

We shall also define the triangulated torus with a hole graph. This can be obtained from a triangulated torus by deleting those vertices of a triangulated disc subgraph that are not on its boundary.

3 Triangulations of the double torus

The proof method of Nevo and Tarabykin is a constructive characterisation. In all of the irreducible triangulations of the surfaces they investigated they found a spanning planar Laman graph. Then they proved that the vertex splits that generate the larger triangulations also maintain the existence of these spanning planar Laman subgraphs. This method however does not seem to be usable for other surfaces as the number of their irreducible triangulations is too large.

Theorem 8 *Let G be a triangulation of the double torus. Then G contains a spanning planar Laman subgraph.*

Proof: Suppose for a contradiction that the statement of the theorem is false. Let G be a triangulation of the double torus which is a counterexample. Choose G in such a way that $|V(G)|$ is as small as possible among the counterexamples.

By Theorem 3 there is an NSC K in G. Take an embedding of G into the double torus in such a way that K is an NSC. If we cut the double torus along the embedding of K we get two tori with one hole each. Let G_1 and G_2 be the two subgraphs of G that are spanned by the vertices that lie on these two tori with hole. The vertices of K are vertices of both G_1 and G_2. We can extend G_1 and G_2 into two triangulations of the torus by adding edges between some pairs of vertices in $V(K)$. Let these graphs be H_1 and H_2.

Now we will show that H_1 has a spanning subgraph T_1 which is a triangulation of the cylinder and contains every edge in $E(K)$. To see this delete the smallest possible subset of edges $F \subseteq E(H_1) - E(K)$ such that $H_1 - F = T$ is planar. We claim that T is a subgraph with the desired properties.

T is clearly spanning and planar and contains every edge in $E(K)$. What we have to show is that T is a triangulation of the cylinder. It is also easy to see that T does not have a cut-edge. Suppose that e is a cut-edge in T. In this case we can add back any edge between the two components of $T - e$, thus U is not minimal, a contradiction. We can then conclude that the boundary of every face is a cycle. What is left to see is that there are at most two faces that are not a triangle. Indeed, if there are more than two larger faces, we can easily get a contradiction using the Euler characteristics of the torus. Then there is a pair of non-triangle faces such that no edge can run between them. But this contradicts the minimality of U.

Let $J_1 = G_1 - U$ be a planar spanning subgraph of G_1. J_2, a spanning planar subgraph of G_2 can be defined similarly. Now merge J_1 and J_2 along K, and let this graph be J. It is easy to see that J is also planar, as K is a boundary of a face in both J_1 and J_2.

As J is a spanning planar subgraph of G, if J is also rigid, then the proof is complete. Suppose now that J is not rigid. It is only possible, if at least one of J_1 and J_2 is not rigid by Lemma 7. Now suppose

that J_1 is rigid and J_2 is not. Then J must also be rigid as T_2 is rigid and every edge in $E(J_2) \setminus E(T_2)$ is implied by J_1. Thus we conclude that none of J_1 and J_2 is rigid. By Lemma 6 this is only possible if the boundaries of the three non-triangle faces are not disjoint. Namely, if there are edges in $E(K)$ that are not in a triangle face in J.

Let $xy \in E(K)$ be such an edge, and let A_1 denote the cycle on the boundary of the non-triangle face for which $xy \in E(A_1)$. xy is in a triangle in G_1 thus the third vertex z of this triangle must be on the third non-triangle face with boundary B_1. Moreover, by the minimality of U every neighbour of x and y is on B_1. Thus xy is not contained in a non-facial triangle in G_1. The same is true for xy in G_2. Thus the edge xy is contractible in G.

Let G' be the graph that we get by contracting xy in G. By the minimality of G G' is not a counterexample and so has a spanning planar Laman subgraph. But then performing the vertex split on G' which is the reverse of the contraction of xy shows that G also has the desired spanning subgraph. This contradiction completes the proof. □

Now we can state the main result regarding the rigidity of triangulated double torus graphs. It follows from Theorems 5 and 8.

Theorem 9 *Let $G = (V, E)$ a triangulation of the double torus. Then if $A \subseteq \mathbb{R}^2$ is a generic set with $|A| = 26$ there is an infinitesimally rigid realization $p : V \to A$ of G.*

4 Other surfaces, future work

By Theorem 4 it is known that there is an NSC in every triangulation of the Klein bottle, the triple cross surface or the quadruple cross surface. Moreover, Sulanke also proved the following:

Theorem 10 *[11] Every triangulation of the quadruple cross surface has an NSC which separates the surface into two surfaces each with genus 2. Every triangulation of the quadruple cross surface has an NSC which separates the surface into two surfaces with genus 1 and 3, respectively.*

Theorems 4 and 10 can be starting points to prove the corresponding result for the triple cross surface or the quadruple cross surface. As these contain an NSC that separates them into two surfaces with holes with genus at most two, one could try to use the spanning planar Laman subgraphs that exist in them to show that the original surface also has one such subgraph.

References

[1] D. Barnette, Generating the triangulations of the projective plane, *J. Comb. Theory, Ser. B* **33** (1982)

[2] Zs. Fekete, T. Jordán, Rigid realizations of graphs on small grids, *Comput. Geom.* **32** (2005)

[3] A. Fogelsanger, The generic rigidity of minimal cycles, *PhD thesis, Cornell University, Ithaca* (1988)

[4] D. L. G. Jennings, Separating cycles in triangulations of the double torus, *Ph.D. thesis, Vanderbilt University* (2003)

[5] Cs. Király, Rigid realizations of graphs with few locations in the plane, *European J. Combin.* **94** (2021)

[6] G. Laman, On graphs and rigidity of plane skeletal structures, *J. Engineering Mathematics* **4** (1970)

[7] S. Lavrenchenko, Irreducible triangulations of the torus, *Journal of Soviet Mathematics* **51** (1990)

[8] S. Lawrencenko and S. Negami, Irreducible triangulations of the klein bottle, *J. Comb. Theory, Ser. B* **70** (1997)

[9] E. Nevo, S. Tarabykin, Vertex spanning planar Laman graphs in triangulated surfaces, *arXiv:2205.00558* (2022)

[10] M. Sitharam, A. St. John, J. Sidman (Eds.), Handbook of Geometric Constraint Systems Principles, *Chapman and Hall/CRC* (2017)

[11] T. Sulanke, Irreducible triangulations of low genus surfaces, *arXiv:math/0606690* (2006)

[12] T. Sulanke, Note on the irreducible triangulations of the klein bottle, *J. Comb. Theory, Ser. B* **96** (2006)

Extremal graphs without long paths and large cliques

Gyula O.H. Katona

MTA Rényi Institute, Budapest, Hungary
ohkatona@renyi.hu

Chuanqi Xiao

Central European University, Budapest, Hungary
chuanqixm@gmail.com

Abstract: Let $\mathcal{F}$ be a family of graphs. A graph is called $\mathcal{F}$-free if it does not contain any member of $\mathcal{F}$ as a subgraph. The Turán number of $\mathcal{F}$ is the maximum number of edges in an n-vertex $\mathcal{F}$-free graph and is denoted by $\mathrm{ex}\big(n,\mathcal{F}\big)$. The same maximum under the additional condition that the graphs are connected is $\mathrm{ex}_{\mathrm{conn}}\big(n,\mathcal{F}\big)$. Let P_k be the path on k vertices, K_m be the clique on m vertices. We determine $\mathrm{ex}(n,\{P_k,K_m\})$ if $k>2m-1$ and $\mathrm{ex}_{\mathrm{conn}}(n,\{P_k,K_m\})$ if $k>m$ for sufficiently large n.

Extremal graph, Turán type theorem

1 Introduction

In the present paper, all graphs considered are undirected, finite and contain neither loops nor multiple edges. Let G be such a graph, the vertex and edge sets of G are denoted by $V(G)$ and $E(G)$, the numbers of vertices and edges in G by $v(G)$ and $e(G)$, respectively. We denote the degree of a vertex v in G by $d_G(v)$, the neighborhood of the vertex set V in G by $N_G(V)$. Let U_1, U_2 be vertex sets, denote by $e_G(U_1,U_2)$ the number of edges between U_1 and U_2 in G. We write $d(v)$ instead of $d_G(v)$, $N(V)$ instead of $N_G(V)$ and $e(U_1,U_2)$ instead of $e_G(U_1,U_2)$ if the underlying graph G is unambiguous. Denote by I_n the independent set on n vertices, by $G[B]$ the subgraph of G induced by the vertex set B and by $\overline{G}$ the edge complement of the graph G. A component of an undirected graph is an induced subgraph in which any two vertices are connected to each other by paths, and which is connected to no additional vertices in the rest of the graph. A vertex v in a graph G is called a cut vertex if deleting v from G increases the number of components of G.

Let $\mathcal{F}$ be a family of graphs. A graph is called $\mathcal{F}$-free if it does not contain any member of $\mathcal{F}$ as a subgraph. The Turán number of $\mathcal{F}$ is the maximum number of edges in an n-vertex $\mathcal{F}$-free graph and is denoted by $\mathrm{ex}\big(n,\mathcal{F}\big)$. Denote by $\mathrm{EX}(n,\mathcal{F})$ the set of $\mathcal{F}$-free graphs on n vertices with $\mathrm{ex}(n,\mathcal{F})$ edges and call a graph in $\mathrm{EX}(n,\mathcal{F})$ an extremal graph for $\mathcal{F}$. Let P_k be the path on k vertices, K_m be the clique on m vertices.

Vertices u and v are connected if there exists a path from u to v. Two disjoint vertex sets U and W are completely joined in G if $uw\in E(G)$ for all $u\in U$ and $w\in W$. Denote by $G_1\bigotimes G_2$ the graph obtained from $G_1\cup G_2$, the vertex disjoint union of graphs G_1 and G_2, and completely join $V(G_1)$ and $V(G_2)$. The Turán graph $T(n,p)$ is a complete multipartite graph formed by partitioning a set of n vertices into p subsets, with sizes as equal as possible, and connecting two vertices by an edge if and only if they belong to different subsets. Denote its size by $t(n,p)$.

In 1941, Turán [5] determined the Turán number for p-clique.

Theorem 1 (Turán[5]) *The number of edges in an n-vertex K_p-free ($p\geq 3$) graph is at most $t(n,p-1)$. Furthermore, $T(n,p-1)$ is the unique extremal graph.*

In 1959, Erdős and Gallai [2] determined the Turán number for P_k.

Theorem 2 (Erdős and Gallai[2]) *Let G be an n-vertex graph with more than $\frac{(k-2)n}{2}$ edges, $k \geq 2$. Then G contains a copy of P_k.*

Faudree and Schelp[3] and independently Kopylov [4] improved this result determining $\mathrm{ex}(n, P_k)$ for every $n > k > 0$ as well as the corresponding extremal graphs.

Theorem 3 (Faudree and Schelp[3] and independently Kopylov [4]) *Let $n \equiv r \pmod{k-1}$, $0 \leq r \leq k-1$, $k \geq 2$. Then*

$$\mathrm{ex}(n, P_k) = \frac{1}{2}(k-2)n - \frac{1}{2}r(k-1-r).$$

Faudree and Schelp also described the extremal graphs which are either

(a) vertex disjoint union of m $(n = m(k-1)+r)$ complete graphs K_{k-1} and a K_r or

(b) k is even and $r = \frac{k}{2}$ or $\frac{k}{2}-1$ then another extremal graph can be obtained by taking a vertex disjoint union if t copies of K_{k-1} $(0 \leq t \leq m)$ and a copy of $K_{\frac{k}{2}-1} \bigotimes \overline{K}_{n-(t+\frac{1}{2})(k-1)+\frac{1}{2}}$.

Kopylov[4] considered the extremal problem for P_k taken over all connected graphs. He determined the extremal values, 30 years later Balister, Győri, Lehel and Schelp found all the extremal graphs, too.

Theorem 4 (Balister, Győri, Lehel and Schelp[1]) *Let G be a connected graph on n vertices containing no path on k vertices, $n > k \geq 4$. Then $e(G)$ is bounded above by the maximum of $\binom{k-2}{2} + (n-k+2)$ and $\binom{\lceil \frac{k}{2} \rceil}{2} + \lfloor \frac{k-2}{2} \rfloor (n - \lceil \frac{k}{2} \rfloor)$. If equality occurs then G is either $\left(K_{k-3} \cup \overline{K}_{n-k+2}\right) \bigotimes K_1$ or $\left(K_{k-2\lfloor \frac{k}{2} \rfloor + 1} \cup \overline{K}_{n-\lceil \frac{k}{2} \rceil}\right) \bigotimes K_{\lfloor \frac{k}{2} \rfloor - 1}$.*

2 Main result

Now let us turn to the problem of the present paper: try to determine $\mathrm{ex}(n, \{P_k, K_m\})$. If $k \leq m$ then this is simply $\mathrm{ex}(n, P_k)$, therefore we can suppose $k > m$ for the rest of the paper.

Construction 1: Suppose $\lfloor \frac{k}{2} \rfloor - 1 \leq n$. $G_1 = T(\lfloor \frac{k}{2} \rfloor - 1, m-2) \bigotimes \overline{K_{n-\lfloor \frac{k}{2} \rfloor + 1}}$.
The number of the edges in this graph is

$$f_n(m,k) = \left(\left\lfloor \frac{k}{2} \right\rfloor - 1\right)\left(n - \left\lfloor \frac{k}{2} \right\rfloor + 1\right) + t\left(\left\lfloor \frac{k}{2} \right\rfloor - 1, m-2\right).$$

Construction 2: Suppose $k-1|n$, let $G_2 = \frac{n}{k-1}T(k-1, m-1)$ denote the graph obtained by taking $\frac{n}{k-1}$ vertex-disjoint copies of $T(k-1, m-1)$.
Clearly, the graphs $T(\lfloor \frac{k}{2} \rfloor - 1, m-2) \bigotimes \overline{K_{n-\lfloor \frac{k}{2} \rfloor + 1}}$ and $\frac{n}{k-1}T(k-1, m-1)$ are $\{K_m, P_k\}$-free.

We believe that for large n (number of vertices) one of these constructions maximize the number of edges under the assumption that the graph contains neither a K_m nor a P_k. More precisely we guess that either Construction 1 gives the largest number of edges or the maximum is between $\frac{n}{k-1}t(k-1, m-1) - c(k,m)$ and $\frac{n}{k-1}t(k-1, m-1)$ where $c(k,m)$ does not depend on n.

But we are able to prove only the following two theorems.

Theorem 5 *Let G be a connected n-vertex $\{K_m, P_k\}$-free graph $m < k$. For sufficiently large n $(> N(k))$,*

$$\mathrm{ex}_{\mathrm{conn}}(n, \{K_m, P_k\}) = \left(\left\lfloor \frac{k}{2} \right\rfloor - 1\right)n + t\left(\left\lfloor \frac{k}{2} \right\rfloor - 1, m-2\right) - \left(\left\lfloor \frac{k}{2} \right\rfloor - 1\right)^2,$$

that is, Construction 1 is an extremal graph.

Theorem 6 *Let G be an n-vertex $\{K_m, P_k\}$-free graph, $2m-1 < k$. For sufficiently large n $(> N'(k))$,*

$$\mathrm{ex}(n, \{K_m, P_k\}) = \left(\left\lfloor \frac{k}{2} \right\rfloor - 1\right)n + t\left(\left\lfloor \frac{k}{2} \right\rfloor - 1, m-2\right) - \left(\left\lfloor \frac{k}{2} \right\rfloor - 1\right)^2,$$

that is, Construction 1 is an extremal graph.

References

[1] P. N. Balister, E. Győri, J. Lehel and R. H. Schelp, Connected graphs without long paths, *Discrete Math.* **308** (2008), 4487–4494.

[2] P. Erdős and T. Gallai, On maximal paths and circuits of graphs, *Acta Math. Acad. Sci. Hungar.* **10** (1959), 337-356.

[3] R. J. Faudree and R. H. Schelp, Path Ramsey numbers in multicolorings, *J. Combin. Theory. Ser. B* **19** (1975), 150–160.

[4] G. N. Kopylov, Maximal paths and cycles in a graph, *Dokl. Akad. Nauk SSSR* **234(1)** (1977) 19 – 21. (English translation: *Soviet Math. Dokl.* **18(3)** (1977), 593–596).

[5] P. Turán, On an extremal problem in graph theory (in Hungarian), *Mat. és Fiz. Lapok* **48** (1941), 436–452.

Orientation of good covers

Péter Ágoston[1]

ELTE Eötvös Loránd University,
Budapest, Hungary
agostonp95@gmail.com

Gábor Damásdi[2]

ELTE Eötvös Loránd University,
Budapest, Hungary
gabor.damasdi@gmail.com

Balázs Keszegh[1][3][4]

Alfréd Rényi Institute of Mathematics and
ELTE Eötvös Loránd University,
Budapest, Hungary
keszegh@renyi.hu

Dömötör Pálvölgyi[1][3]

ELTE Eötvös Loránd University,
Budapest, Hungary
domotor.palvolgyi@ttk.elte.hu

Abstract: We study systems of orientations on triples that satisfy the following so-called interiority condition: $\circlearrowleft(ABD) = \circlearrowleft(BCD) = \circlearrowleft(CAD) = 1$ implies $\circlearrowleft(ABC) = 1$ for any A, B, C, D. We call such an orientation a P3O (partial 3-order), a natural generalization of a poset, that has several interesting special cases. For example, the order type of a planar point set (that can have collinear triples) is a P3O; we denote a P3O realizable by points as p-P3O.

If we do not allow $\circlearrowleft(ABC) = 0$, we obtain a T3O (total 3-order). Contrary to linear orders, a T3O can have a rich structure. A T3O realizable by points, a p-T3O, is the order type of a point set in general position.

In [1] we defined a 3-order on pairwise intersecting convex sets; such a P3O is called a C-P3O. In this paper we extend this 3-order to pairwise intersecting good covers; such a P3O is called a GC-P3O. If we do not allow $\circlearrowleft(ABC) = 0$, we obtain a C-T3O and a GC-T3O, respectively.

The main result of this paper is that there is a p-T3O that is not a GC-T3O, implying also that it is not a C-T3O—this latter problem was left open in our earlier paper. Our proof involves several combinatorial and geometric observations that can be of independent interest. Along the way, we define several further special families of GC-T3O's.

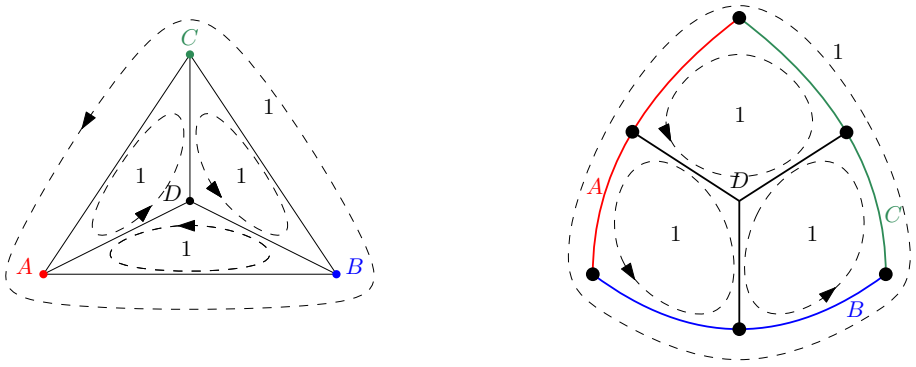
Keywords: convex set, good cover, orientation

[1]This research has been implemented with the support provided by the Ministry of Innovation and Technology of Hungary from the National Research, Development and Innovation Fund, financed under the ELTE TKP 2021-NKTA-62 funding scheme.

[2]Supported by the ÚNKP-21-3 New National Excellence Program of the Ministry for Innovation and Technology from the source of the National Research, Development and Innovation fund.

[3]Supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences and by the ÚNKP-21-5 and ÚNKP-22-5 New National Excellence Program of the Ministry for Innovation and Technology from the source of the National Research, Development and Innovation Fund.

[4]Supported by the National Research, Development and Innovation Office – NKFIH under the grant K 132696 and FK 132060.



(a) (b)

Figure 1: The interiority condition is true for points (a) and also for pairwise intersecting convex sets (b); here $\circlearrowleft(ABD) = \circlearrowleft(BCD) = \circlearrowleft(CAD) = \circlearrowleft(ABC)$, or equivalently, we can write $D \in conv(ABC)$.

1 Introduction

Given some base set, a mapping $\circlearrowleft$ from its ordered triples to $\{\pm 1, 0\}$ is a *partial orientation* if

$$\circlearrowleft(ABC) = \circlearrowleft(CAB) = \circlearrowleft(BCA) = -\circlearrowleft(ACB) = -\circlearrowleft(BAC) = -\circlearrowleft(CBA) \text{ for every } A, B, C.$$

If $\circlearrowleft$ never takes zero, then $\circlearrowleft$ is a *total orientation*. An orientation satisfies the *interiority condition* if

$$\circlearrowleft(ABD) = \circlearrowleft(BCD) = \circlearrowleft(CAD) = 1 \text{ implies } \circlearrowleft(ABC) = 1 \text{ for every } A, B, C, D.$$

If $\circlearrowleft(ABD) = \circlearrowleft(BCD) = \circlearrowleft(CAD) = 1$ or $\circlearrowleft(ABD) = \circlearrowleft(BCD) = \circlearrowleft(CAD) = -1$ for some A, B, C, D, then we write $D \in conv(ABC)$. See Figure 1(a). (In the definition of $conv(ABC)$ the order of A, B, C is not relevant.)

A total orientation that satisfies the interiority condition is a T3O (total 3-order), and a partial orientation is a P3O (partial 3-order). The notion T3O was introduced by Knuth [5] under the name *interior triple system*, according to Knuth "for want of a better name." He noted that taking the orientations (in the well-known geometric sense) of all triples of a planar point set in general position we get a T3O, while if we allow collinearity, we get a P3O. The equivalence classes of point sets giving the same P3O are called the order types [3], a notion having a broad literature. We say that a T3O (resp. P3O) that has a realization by a planar set of points in this way is a p-T3O (resp. p-P3O). We denote the family of all P3O's by $\mathcal{P}3\mathcal{O}$ and, similarly, for its subfamilies, we use the calligraphic $\mathcal{T}3\mathcal{O}$, p-$\mathcal{P}3\mathcal{O}$, p-$\mathcal{T}3\mathcal{O}$, respectively.

Note that the definition of a P3O is similar to the definition of partially ordered sets, therefore our choice for its name. Indeed, a poset is a mapping from the ordered pairs of its base sets to $\{\pm 1, 0\}$ requiring antisymmetry and transitivity. Similarly, a P3O does the same for ordered triples, but in our case requiring the interiority condition.

In a companion paper [1], motivated by a lemma of Jobson et al. [4] (see also Lehel and Tóth [6]), we have defined an orientation on intersecting planar convex sets, as follows. If $A \cap B \cap C \neq \emptyset$, then $\circlearrowleft(ABC) = 0$. Otherwise, by [4], $\mathbb{R}^2 \setminus (A \cup B \cup C)$ has one bounded component, and its boundary has exactly one arc from each of the boundaries of A, B and C. We defined $\circlearrowleft(ABC) = 1$ if in cyclic counterclockwise order these arcs belong to A, B, C, and proved that $\circlearrowleft$ satisfies the interiority condition, i.e., it is a 3-order[1]. Denote the subfamily of $\mathcal{T}3\mathcal{O}$ and $\mathcal{P}3\mathcal{O}$ that have a realization by pairwise intersecting planar convex sets by C-$\mathcal{T}3\mathcal{O}$ and C-$\mathcal{P}3\mathcal{O}$, respectively. In particular, if no three sets from a pairwise

[1]There is also a quite different definition of orientation of triples when the convex sets are pairwise disjoint, for its short discussion and further references see the respective remark in [1].

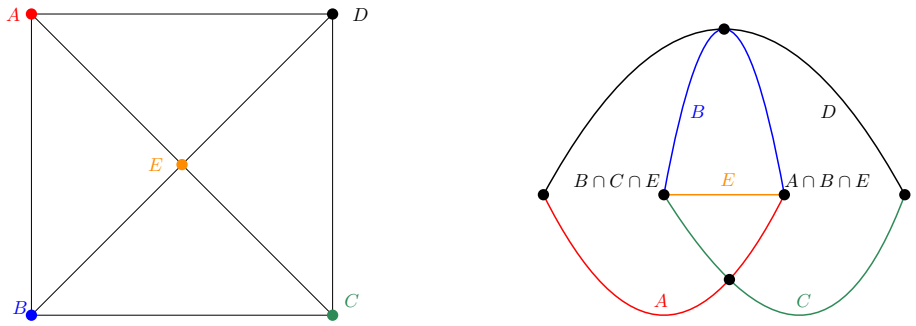


Figure 2: A p-P3O, realized on the left by the four vertices of a square (A, B, C, D) and their center (E), that was shown in [1] not to be a C-P3O, but because of the above realization on the right by a good cover it is a GC-P3O.

intersecting convex family have a common point (called a *holey family* in [1]), the orientation $\circlearrowleft$ gives a C-T3O on them. In this paper, we extend the orientation $\circlearrowleft$ to good covers, and denote the respective families by GC-$\mathcal{T}3\mathcal{O}$ and GC-$\mathcal{P}3\mathcal{O}$. A family of sets is a *good cover* if the intersection of any subfamily is contractible or empty [7].

As convex sets are always good covers, C-$\mathcal{P}3\mathcal{O}$ $\subset$ GC-$\mathcal{P}3\mathcal{O}$ $\subset$ $\mathcal{P}3\mathcal{O}$. Both inequalities are strict: GC-$\mathcal{P}3\mathcal{O}$ $\neq$ $\mathcal{P}3\mathcal{O}$ follows from Theorem 1, while C-$\mathcal{P}3\mathcal{O}$ $\neq$ GC-$\mathcal{P}3\mathcal{O}$ because with good covers a certain five-point configuration can be realized whose p-P3O was shown not be a C-P3O in [1]; see Figure 2.

We have shown in [1] that C-$\mathcal{T}3\mathcal{O}$ $\subsetneq$ p-$\mathcal{T}3\mathcal{O}$, i.e. that C-$\mathcal{T}3\mathcal{O}$ is a proper subfamily of p-$\mathcal{T}3\mathcal{O}$ (implying also C-$\mathcal{P}3\mathcal{O}$ $\subsetneq$ p-$\mathcal{P}3\mathcal{O}$), and p-$\mathcal{P}3\mathcal{O}$ $\subsetneq$ C-$\mathcal{P}3\mathcal{O}$. In this paper we establish the strengthening p-$\mathcal{T}3\mathcal{O}$ $\subsetneq$ C-$\mathcal{T}3\mathcal{O}$, i.e., that there is a 3-order that is realizable by points in general position, but not by pairwise intersecting convex sets. This follows from the following more general result.

Theorem 1 p-$\mathcal{T}3\mathcal{O}$ $\subsetneq$ GC-$\mathcal{T}3\mathcal{O}$.

Our proof will first establish that the 3-order of some point set is not realizable by some special subfamily of good covers, and then gradually increase the complexity of this subfamily, while also making our point set larger, until we establish the theorem.

In the next Section 2 we define an orientation $\circlearrowleft$ on good covers, and show that in a GC-T3O realization we can assume that the sets are pairwise once intersecting topological trees, which also helps proving that $\circlearrowleft$ satisfies the interiority condition, thus a 3-order. The rest of the proof is omitted due to space constraints and can be found in [2].

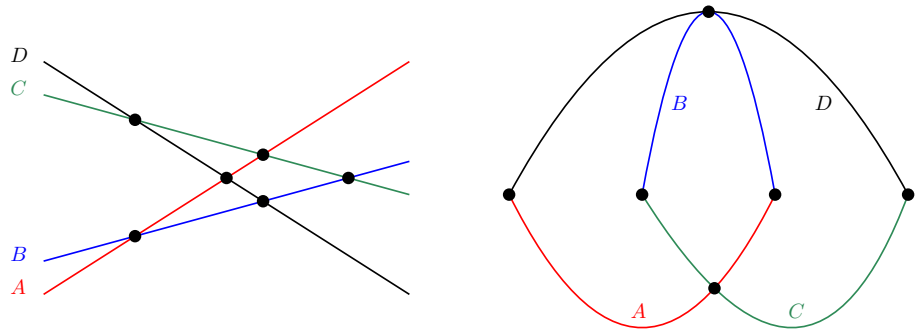
2 Good covers and topological trees

Let us repeat our main definition: a family of sets is a *good cover* if the intersection of any subfamily is contractible or empty [7]. For example, any family of convex sets is a good cover. Another example is any family of sets that pairwise intersect in at most one point, which can either be a crossing point or a tangency.[2]

Given three pairwise intersecting sets, A, B, C, define $\circlearrowleft(A, B, C)$ as follows. If $A \cap B \cap C \neq \emptyset$, then $\circlearrowleft(A, B, C) = 0$. Otherwise, if there is a Jordan curve γ such that γ is the concatenation of three curves, γ_A, γ_B, γ_C, such that $\gamma_A \subset A$, $\gamma_B \subset B$ and $\gamma_C \subset C$, then if $\gamma_A, \gamma_B, \gamma_C$ follow each other around γ in counterclockwise order, define $\circlearrowleft(A, B, C) = 1$, while if $\gamma_A, \gamma_B, \gamma_C$ follow each other around γ in clockwise order, define $\circlearrowleft(A, B, C) = -1$.[3] We will show that for good covers with pairwise intersecting sets the

[2] Given two sets that intersect once in some neighborhood, their intersection point is called a tangency, or touching point, if it can be eliminated by a small perturbation of the sets.

[3] Note that for convex sets this definition coincides with the one from [1].



(a) (b)

Figure 3: Two good covers that consist of pairwise intersecting topological trees. The four sets depicted in each figure are such that no point is contained in three sets, so their orientation is a GC-T3O, and no three sets satisfy the premise of the interiority condition: $\circlearrowleft(ABC) = \circlearrowleft(BCD) = \circlearrowleft(CDA) = \circlearrowleft(DAB)$. Note that during the closed walk $A{\cap}B$–$B{\cap}C$–$C{\cap}D$–$D{\cap}A$–$A{\cap}B$ in (a) we wind around once, while in (b) we wind around twice.

above orientation $\circlearrowleft$ is well-defined (see Claim 2) and satisfies the interiority condition (see Corollary 8), thus it is a 3-order. For an example that satisfies the premise of the interiority condition, see Figure 1(b), while no three of the four sets in Figures 3(a) and (b) satisfy the premise.

First we fix some notations. We make no distinction between an element and the set representing it. The restriction of the orientation $\circlearrowleft$ to some elements $\mathcal{X} = \{X_1, \ldots, X_n\}$ from a family (point sets, good covers, etc.), is denoted by $\circlearrowleft(X_1, \ldots, X_n)$ or simply $\circlearrowleft(\mathcal{X})$. For brevity, when talking about orientations, we may even refer to $\circlearrowleft(\mathcal{X})$ simply as $\mathcal{X}$ if it leads to no confusion.

Claim 2 *The orientation $\circlearrowleft$ is well-defined.*

This simple topological claim is probably already known, its proof can be found also in the full version of the paper.

We define a special subfamily of good covers, where each set is a topological tree.

A *topological tree* is an injective embedding of a (graph theoretic) tree, that has no degree two vertices, into the plane, such that vertices are mapped to points, and edges are mapped to simple curves. The images of the degree one vertices are called *leaves*, while the images of the vertices with degree at least three are called *branching points*.

A family of topological trees forms a good cover if every pair of trees intersect at most once. (This is not an if and only if condition, but it will be more convenient for us to work only with such families.) For two trees A and B we denote their intersection point by $A{\cap}B$, i.e., $A \cap B = \{A{\cap}B\}$.[4] For three trees, A, B and C, we have $\circlearrowleft(A, B, C) = 0$ if and only if their pairwise intersection points coincide.

We need to introduce some notation. See Figure 4 for illustration of the next definition.

Definition 3 *Suppose that X is a topological tree and the point p_i is in X for each $1 \le i \le k$.*

Define $X[p_1\text{-}\ldots\text{-}p_k]$ to be the minimal connected subset of X which contains p_i for every i. In particular, $X[p_1\text{-}p_2]$ is the path connecting p_1 and p_2 in X. Note that $X[p_1\text{-}\ldots\text{-}p_k] = \cup_{i,j} X[p_i\text{-}p_j]$.

Define $X[p_1, \ldots, p_k]$ to be the minimal connected subset X' of X which contains p_i for every i, and for which every connected component of $X \setminus X'$ has p_i on its boundary for some i.

Define $X[p_1|p_2]$ to be the set of those points $p \in X$ for which $X[p\text{-}p_2]$ contains p_1.

If $A_1, \ldots A_k$ are topological trees that intersect X once and $p_i = X{\cap}A_i$, then for brevity we can replace p_i in the above notations with A_i. For example, $X[A_1\text{-}A_2] = X[p_1\text{-}A_2] = X[p_1\text{-}p_2]$.

[4]The difference is that there is less space around the new math operator. The similarity can lead to no confusion, as these two operators denote practically the same thing.

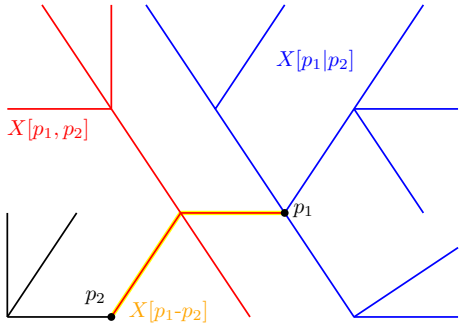


Figure 4: Parts of the topological tree X.

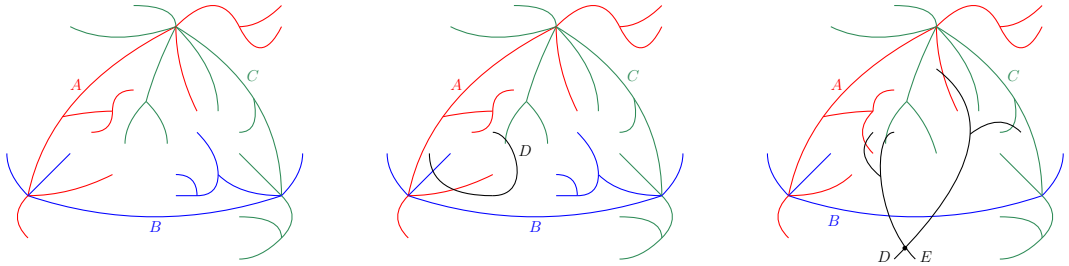


Figure 5: $\text{人}(ABC)$ and the hairs.

Now we are ready to prove our main structural tool.

Proposition 4 *Assume that in a planar family:*

§1 Every set is a topological tree.

§2 Every pair of sets intersects in exactly one point.[5]

Then the following hold:

§3 The union of any three sets, A, B, C, without a common point, contains exactly one cycle, i.e., a Jordan curve. The interior of this cycle is called the *hollow* and is denoted by $\text{人}(ABC)$.[6] The boundary of $\text{人}(ABC)$ consists of parts of $A, A\cap B, B, B\cap C, C, C\cap A$, in this order, if $\circlearrowleft(ABC) = 1$. From the boundary of $\text{人}(ABC)$, there can be subtrees of A, B, C going inwards and outwards, these we call *hairs*. Hairs might have branchings on them but they are disjoint from each other. See Figure 5(a).

§4 Any four sets satisfy the interiority condition, thus $\circlearrowleft$ is a 3-order.
Further, if $D \in conv(ABC)$, then

$$\text{人}(ABD)\dot\cup\text{人}(BCD)\dot\cup\text{人}(ACD)\dot\cup D[A\text{-}B\text{-}C] \cup \{\textit{at most one-one hairpart of } A, B \textit{ and } C\}$$

gives a partition of $\text{人}(ABC) \cup \{A\cap D, B\cap D, C\cap D\}$. In particular, $D[A, B, C] \subset \text{人}(ABC)$, apart from its endpoints, $\{A\cap D, B\cap D, C\cap D\}$.

§5 If $D, E \in conv(ABC)$ and the orientation on A, B, C, D, E is realizable by five points in general position, then $D\cap E \in \text{人}(ABC)$.

By §3 and §4, the orientation $\circlearrowleft$ defined for good covers gives a P3O for any family satisfying the conditions §1 and §2. We call a P3O that is realizable this way a Tr-P3O. If in addition no three trees have a common intersection, then it is also called a Tr-T3O representation.

[5] Two trees are allowed to have multiple branches from their intersection point.
[6] Note that for convex sets this definition coincides with the one from [1].

Remark 5 *Note that because of the hairs several intuitive statements are false. For example, it is possible that* $D[A,B,C] \subset \text{人}(ABC)$ *but* $D \notin conv(ABC)$*. See Figure 5(b). Also, if in §5 we only assume that* $D, E \in conv(ABC)$*, then it is possible that* $D \cap E \notin \text{人}(ABC)$*. See Figure 5(c).*

PROOF:[Proof of §3.] As A is a tree, there is exactly one path in A between $A \cap B$ and $A \cap C$, and this cannot intersect B or C. Similarly, there is one path in B between $A \cap B$ and $B \cap C$, and there is one path in C between $A \cap C$ and $B \cap C$. The union of these three paths gives the required Jordan curve. □

PROOF:[Proof of §4.]

Assume that $\circlearrowleft(ABC) = \circlearrowleft(ABD) = \circlearrowleft(BCD) = \circlearrowleft(CAD) = 1$. To show that the interiority condition holds we need to prove that $\circlearrowleft(ABC) = 1$. First, we assume $\circlearrowleft(ABC) \neq 0$, i.e., $A \cap B \cap C = \emptyset$.

Consider $D' = D[A\text{-}B\text{-}C]$. Either D' is a path, or a Y-shaped star.

In the former case, we can assume without loss of generality that $B \cap D \in D[A\text{-}C] = D'$, i.e., $B \cap D$ lies between $A \cap D$ and $C \cap D$ on D. From $\circlearrowleft(CAD) = 1$, we know on which side of D' the hollow $\text{人}(ACD)$ lies. See Figure 6(a).

Note that $\text{人}(ABD) \subset \text{人}(ACD)$ would imply $\circlearrowleft(ABD) = \circlearrowleft(ACD) \neq \circlearrowleft(CAD)$, contradicting our assumptions. Similarly, $\text{人}(BCD) \not\subset \text{人}(ACD)$.

This implies that none of the paths $B[D\text{-}A]$ and $B[D\text{-}C]$ can start from $B \cap D$ towards the interior of $\text{人}(ACD)$, as otherwise they would need to intersect $\partial\text{人}(ACD)$, either in A or in C, which would give $\text{人}(ABD) \subset \text{人}(ACD)$ or $\text{人}(BCD) \subset \text{人}(ACD)$, respectively, contradicting our previous observation. See Figure 6(a). As both paths start from $B \cap D$ towards the exterior of $\text{人}(ACD)$, we can pretend that D' is a (degenerate) Y-shape such that its leaves in the counterclockwise order are $A \cap D, B \cap D, C \cap D$.

The same argument rules out the possibility that D' is a Y-shape such that its leaves in the counterclockwise order are $A \cap D, C \cap D, B \cap D$.

Therefore, we can conclude that D' needs to be a (possibly degenerate) Y-shape such that its leaves in the counterclockwise order are $A \cap D, B \cap D, C \cap D$. Denote the branching point of D' by D_y (where $D_y = B \cap D$ if D' is degenerate). See Figure 6(b).

Denote the branching point of $A[B\text{-}C\text{-}D]$ by A_y, or if $A[B\text{-}C\text{-}D]$ is a path, then let A_y stand for whichever of $A \cap B, A \cap C$ and $A \cap D$ lies in the middle of the path. In other words, A_y is the point up to which $A[D\text{-}B]$ and $A[D\text{-}C]$ follow the same route starting from $A \cap D$. In particular, $A[D\text{-}A_y]$ does not contain $A \cap B$ or $A \cap C$ in its interior. We similarly define B_y and C_y.

Now, walk along $\partial\text{人}(ABD), \partial\text{人}(BCD)$ and $\partial\text{人}(ACD)$, starting always from D_y. Note that in the union of these three walks, During the walk around $\partial\text{人}(ABD)$, we cover the part from D_y to A_y, then we go from A_y to B_y, then back to D_y. Around $\partial\text{人}(BCD)$, we cover the part from D_y to B_y, then we go from B_y to C_y, then back to D_y. Finally, around $\partial\text{人}(ACD)$, we cover the part from D_y to C_y, then we go

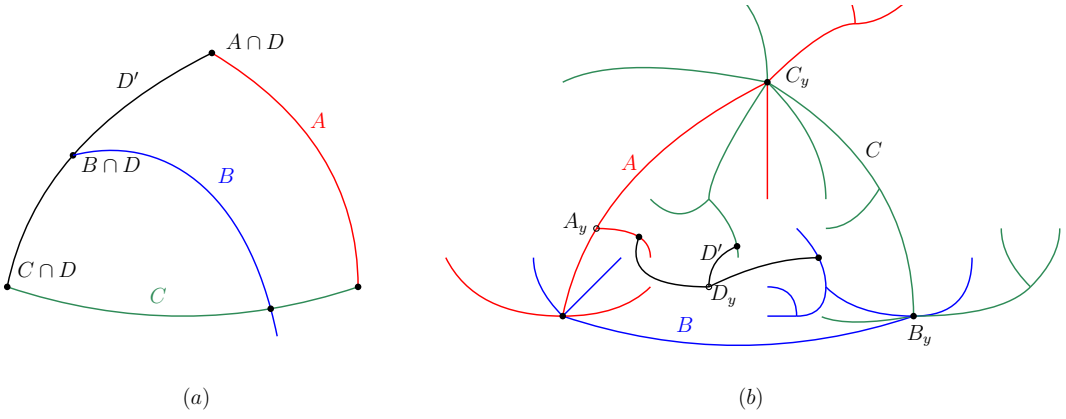
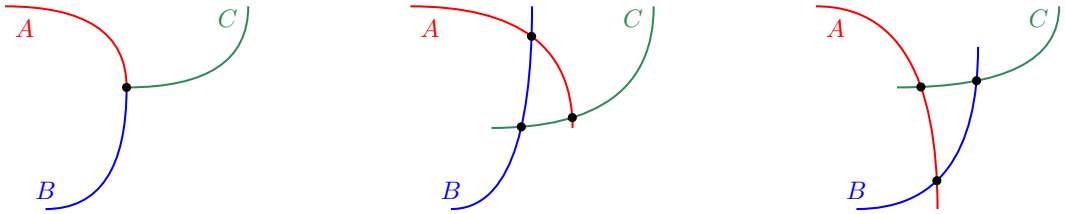


Figure 6: Proof of §4.



(a) (b) (c)

Figure 7: A triple intersection can be perturbed in two ways.

from C_y to A_y, then back to D_y. Note that each part from D' occurs there and back, while all the other parts are disjoint, apart from their endpoints. But this means that by eliminating the there-and-back parts, we get a walk from A_y to B_y to C_y, then back to A_y, that has the same orientation as the original walks, and contains AB, BC and AC. That is, $\circlearrowleft(ABC) = 1$, as claimed. This finishes the proof of the interiority condition if $\circlearrowleft(ABC) \neq 0$.

Now assume for a contradiction that $\circlearrowleft(ABC) = 0$, i.e., $A \cap B \cap C \neq \emptyset$. By §2, $A \cap B \cap C$ is a single point; denote it by x. If we replace A with $A[x\text{-}D]$, then $\circlearrowleft(A, B, C, D)$ remains the same. We can similarly replace B and C with $B[x\text{-}D]$ and $C[x\text{-}D]$, respectively. This way each of A, B and C became a curve ending in x. But then by a slight perturbation of the curves in the vicinity of x we can achieve that they pairwise intersect once but in three different points, such that $\circlearrowleft(ABC)$ becomes -1 (see Figure 7).

But this would contradict the interiority condition for $\circlearrowleft(ABC) \neq 0$, which we have already proved. This finishes the proof of the interiority condition if $\circlearrowleft(ABC) = 0$.

The structural description that we have obtained in the $\circlearrowleft(ABC) \neq 0$ case implies that $\text{人}(ABC) = \text{人}(ABD) \dot\cup \text{人}(BCD) \dot\cup \text{人}(ACD) \dot\cup D' \cup \{$the parts from A_y to $A \cap D$, from B_y to $B \cap D$, and from C_y to $C \cap D\}$—this last part gives the three possible hairs. Since D intersects $\text{人}(ABC)$ exactly three times, this also implies $D[A, B, C] \subset \text{人}(ABC)$. $\square$ Proof:[Proof of §5.] Suppose for a contradiction that $x = D \cap E \notin \text{人}(ABC)$.

By §4, $D[A, B, C] \subset \text{人}(ABC)$ and thus x falls in one connected component of $D \setminus D[A, B, C]$ which implies that the three paths $D[E\text{-}A]$, $D[E\text{-}B]$, $D[E\text{-}C]$ all go the same way from x until they reach $\text{人}(ABC)$. Denote their intersection point with $\partial\text{人}(ABC)$ by D_x. Note that D_x is one of $A \cap D, B \cap D, C \cap D$.

Similarly, the three paths $E[D\text{-}A]$, $E[D\text{-}B]$, $E[D\text{-}C]$, all start the same way from x. Denote their intersection point with $\partial\text{人}(ABC)$ by E_x. This time there is no need to make any wise notes.

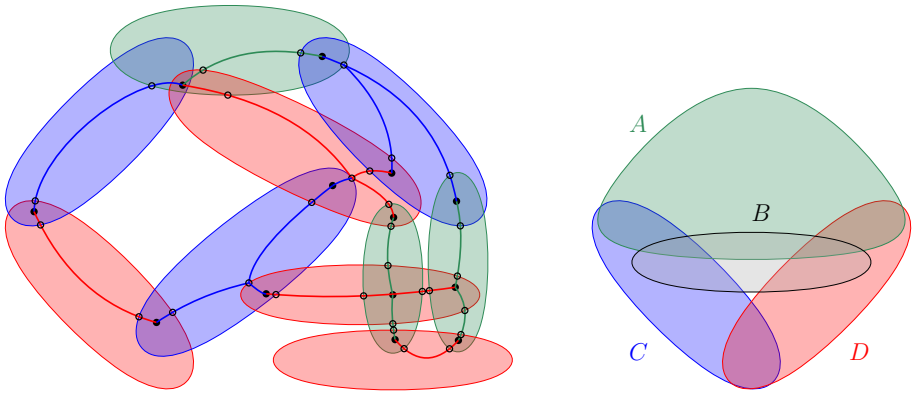
We claim that $\circlearrowleft(ADE) = \circlearrowleft(BDE) = \circlearrowleft(CDE)$. Indeed, this follows from the fact that $\text{人}(ADE)$, $\text{人}(BDE)$ and $\text{人}(CDE)$ are all contained in the union of $\text{人}(ABC)$ and the topological triangle whose vertices are x, D_x, E_x. To see this, note that for any of these hollows, x will be a vertex, while the two sides of the hollow adjacent to x will go through D_x and E_x, respectively, and then continue inside $\text{人}(ABC)$ until they reach the other two vertices of the hollow, because of §4.

But this contradicts that $D, E \in conv(ABC)$ and the orientation on A, B, C, D, E is realizable by five points in general position, as if in this realization the points D and E are contained in the convex hull of A, B and C, then two of these three points will fall on different sides of the DE line, so $\circlearrowleft(ADE) = \circlearrowleft(BDE) = \circlearrowleft(CDE)$ is not possible. $\square$

Next, we prove that $\circlearrowleft$ behaves essentially the same way on any good cover as on topological trees.

Lemma 6 *The sets in any good cover, where at most one triple has a non-empty intersection, can be replaced by topological trees that pairwise intersect at most once, such that $\circlearrowleft$ remains unchanged on all triples.*

Proof: See Figure 8(a) for an illustration of the proof.



(a) (b)

Figure 8: (a) A good cover redrawn with topological trees that pairwise intersect in at most one point. (b) A good cover that cannot be redrawn this way.

We can assume that every set intersects some other set from our family. For each set A from the good cover, and for each connected component A_i of those points that are only in A and in no other set, we do the following. On the boundary of A_i there are pairwise disjoint arcs which are on the boundary of some other set as well. We put a topological star inside A_i whose center is on one of these arcs, and there is a leaf on each of the rest of the arcs.

Now, assume that there is no triple intersection. For each non-empty intersection $A \cap B$, we select a point p_{AB} inside it (p_{AB} will be the intersection point of the topological trees corresponding to A and B), and draw non-crossing curves from p_{AB}, one to every leaf of the earlier defined stars that are on the boundary of $A \cap B$. Each such curve is added to the topological tree it touches at the boundary of $A \cap B$.

If there is a triple intersection $A \cap B \cap C \neq \emptyset$, then we treat $(A \cap B) \cup (A \cap C) \cup (B \cap C)$ as one double intersection, and do the same as before, selecting a point p_{ABC} from $A \cap B \cap C$ (and no other points from $A \cap B$, $A \cap C$, and $B \cap C$, so in this case there are no points p_{AB}, p_{AC}, p_{BC}). In the remainder of the proof, we will not discuss this special triple intersection region in detail—all steps work for it the same way.

It is easy to see that we get topological trees, as all the points inside a set A are eventually connected. Denote this tree by $T_A \subset A$. These topological trees intersect pairwise exactly once, in the point selected inside the intersection of their corresponding sets, i.e., $T_A \cap T_B = p_{AB}$.

We are left to show that the orientations are preserved. As the trees satisfy §2, $\circlearrowleft$ is well-defined on the trees by §3. Take some A, B, C and the (unique) Jordan curve γ such that $\gamma_A \subset T_A$, $\gamma_B \subset T_B$ and $\gamma_C \subset T_C$. As $T_X \subset X$ for any set X, also $\gamma_A \subset A$, $\gamma_B \subset B$ and $\gamma_C \subset C$, so the same γ shows that $\circlearrowleft(T_A T_B T_C) = \circlearrowleft(ABC)$. □

Remark 7 *The condition that at most one triple has a non-empty intersection is necessary because we allow trees to intersect in at most one point. If, for example, $A \cap B \cap C \neq \emptyset$ and $A \cap B \cap D \neq \emptyset$ but $A \cap B \cap C \cap D = \emptyset$, this obviously cannot be realized by trees that pairwise intersect at most once. See Figure 8(b) for such a good cover. But this is essentially the only obstruction—our proof can be modified in a straight-forward way to work also if we require that there are no four sets such that $A \cap B \cap C \neq \emptyset$ and $A \cap B \cap D \neq \emptyset$.*

Corollary 8 *The orientation $\circlearrowleft$ is a 3-order on good covers.*

Proof: We need to show that $\circlearrowleft$ satisfies the interiority condition. Take four sets such that $\circlearrowleft(ABD) = \circlearrowleft(BCD) = \circlearrowleft(CAD) = 1$. In particular, these four sets can have at most one non-

empty triple intersection, $A \cap B \cap C$. Therefore, by Lemma 6 we can convert A, B, C, D to trees while preserving $\circlearrowleft$, and so §4 implies that $\circlearrowleft(ABC) = 1$. $\square$

Remark 9 *This also implies that $\circlearrowleft$ is a 3-order on convex sets, reproving a key lemma from [1].*

Denote a P3O/T3O realizable by good covers (resp. by topological trees), by GC-P3O/GC-T3O (resp. by Tr-P3O/Tr-T3O), and the respective subfamilies by calligraphic, as usual.

Corollary 10 GC-$\mathcal{T}3\mathcal{O}$ = Tr-$\mathcal{T}3\mathcal{O}$.

This implies that to establish Theorem 1, it is enough to prove p-$\mathcal{T}3\mathcal{O} \subsetneq$ Tr-$\mathcal{T}3\mathcal{O}$. However, there is a lot of work left, with multiple intermediate steps, the complete proof is omitted due to space constrains and can be found in [2].

References

[1] P. Ágoston, G. Damásdi, B. Keszegh, and D. Pálvölgyi, Orientation of convex sets, *Preprint, https://arxiv.org/abs/2206.01721*

[2] P. Ágoston, G. Damásdi, B. Keszegh, and D. Pálvölgyi, Orientation of good covers, *Preprint, https://arxiv.org/abs/2206.01723*

[3] Goodman, Jacob E. and Pollack, Richard, Allowable Sequences and Order Types in Discrete and Computational Geometry, *New Trends in Discrete and Computational Geometry, Springer Berlin Heidelberg* **103–134** (1993)

[4] A. S. Jobson, A. E. Kézdy, J. Lehel, T. J. Pervenecki, and G. Tóth., Petruska's question on planar convex sets, *Discrete Mathematics* **343(9):1–13** (2020)

[5] D. E. Knuth., Axioms and hulls, *Lecture Notes in Computer Science* (1992)

[6] J. Lehel and G. Tóth., On the hollow enclosed by convex sets, *Geombinatorics* **30(3):113–122** (2021)

[7] A. Weil, Sur les théorèmes de de Rham, *Commentarii Math. Helv.* **26:119-145** (1952)

A polynomial-time algorithm to compute the toughness of graphs with bounded treewidth

Gyula Y. Katona[1]

Department of Computer Science and Information Theory
Budapest University of technology and Economics
Budapest, Hungary
katona.gyula@vik.bme.hu

Humara Khan

Department of Computer Science and Information Theory
Budapest University of technology and Economics
Budapest, Hungary
humaraawan@gmail.com

Abstract: Let t be a positive real number. A graph is called *t-tough* if the removal of any vertex set S that disconnects the graph leaves at most $|S|/t$ components. The toughness of a graph is the largest t for which the graph is t-tough. We prove that toughness is fixed-parameter tractable parameterized with treewidth. More precisely, we give an algorithm to compute the toughness of a graph G with running time $\mathcal{O}(\mathrm{tw}(G)^{2\mathrm{tw}(G)} \cdot |V(G)|^3)$ where $\mathrm{tw}(G)$ is the treewidth. If the treewidth is bounded by a constant, then this is a polynomial algorithm.

Keywords: toughness, treewidth, fixed-parameter tractable

1 Introduction

1.1 Complexity of toughness

All graphs considered in this paper are finite, simple and undirected. Let $c(G)$ denote the number of components of G. If $S \subseteq V(G)$, then $G - S$ denotes the graph obtained by deleting all vertices of S from G. For a connected graph G, a vertex set $S \subseteq V(G)$ is called a *cutset* if $c(G - S) > 1$.

The notion of toughness was introduced by Chvátal in [11].

Definition 1 *Let t be a real number. A graph G is called* t-tough *if $|S| \geq t \cdot c(G - S)$ for any $S \subseteq V(G)$ with $c(G - S) > 1$. The* toughness *of G, denoted by $\tau(G)$, is the largest t for which G is t-tough, taking $\tau(K_n) = \infty$ for all $n \geq 1$.*

Note that a graph is disconnected if and only if its toughness is 0.

The relation of toughness to Hamiltonian cycles, connectivity and other measures of graph robustness are well researched topics. There are quite a few results on the computational aspects of toughness.

Let t be an arbitrary positive rational number and consider the following problem.

***t*-Tough**
Instance: a graph G.
Question: is it true that $\tau(G) \geq t$?

It is easy to see that for any positive rational number t the problem t-Tough is in coNP: a witness is a vertex set S whose removal disconnects the graph and leaves more than $|S|/t$ components. Bauer et al. proved that this problem is coNP-complete [2] and the problem 1-Tough remains coNP-complete for at least 3-regular graphs [5].

[1]The research reported in this paper was supported by the Higher Education Excellence Program of the Ministry of Human Capacities in the frame of the Artificial Intelligence research area of the Budapest University of Technology and Economics BME FIKP-MI/SC.

Theorem 2 ([2]) *For any positive rational number t, the problem t-TOUGH is coNP-complete.*

Theorem 3 ([5]) *For any fixed integer $r \geq 3$, the problem 1-TOUGH is coNP-complete for r-regular graphs.*

Although the toughness of any bipartite graph (except for the graphs K_1 and K_2) is at most one, the problem 1-TOUGH does not become easier for bipartite graphs [16], and the same is true for smaller t values [14]. In [14], the authors prove similar results for different sub-classes of bipartite graphs. A good survey of further results is [9].

It is widely believed that these complexity results imply that there is no polynomial algorithm that computes the toughness of a given graph even for these special graph classes. On the other hand, for some other graph classes we do have such algorithms. The first example is the class of claw-free graphs. Since the toughness of a claw-free graphs is half of its connectivity [17], and the connectivity of any graph can be computed in polynomial time, so is the toughness of claw-free graphs. Another class of graphs for which this is the case is the class of split graphs. In [16], it was shown that the class of 1-tough graphs can be recognized in polynomial time, and in [20] the same was proved for t-tough split graphs with $t \geq 0$. This was further extended to a larger class, namely to the class of $2K_2$-free graphs [10]. For interval graphs, [15] contains a polynomial algorithm to compute the toughness.

For some special values of t there are also polynomial algorithms to compute the toughness of some regular graphs. For instance, the characterization of 3-regular 3/2-tough graphs in [13] implies that these graphs can be recognized in polynomial time. Some further results are the following.

Theorem 4 ([14]) *For any positive rational number $t < 2/3$, there is a polynomial-time algorithm to recognize t-tough 3-regular graphs.*

Theorem 5 ([14]) *There is a polynomial-time algorithm to recognize 1/2-tough 4-regular graphs.*

1.2 Treewidth

The notion of *treewidth* is proposed in [18, 19] and is a famous parameter in complexity problems. Here we follow the notations of [6]. Given a graph $G = (V, E)$, a tree decomposition of G is a pair $\mathcal{T} = (\mathcal{X}, T)$ where $\mathcal{X} = \{X_i \mid i \in I\} \subseteq \mathcal{P}(V)$ and $T = (I, F)$ is a tree satisfying:

1. $\cup_{i \in I} X_i = V$,
2. for all edges $\{u, v\} \in E$, there exists $i \in I$ such that $\{u, v\} \subseteq X_i$,
3. for all $i, j, k \in I$, if j is on the path from i to k in T, then $X_i \cap X_k \subseteq X_j$.

The elements of set $\mathcal{X}$ which correspond to the nodes of tree T are called *bags*. Note that the last condition can be substituted by the following equivalent condition:

3. for each $v \in V$, the set of nodes $\{i \in I \mid v \in X_i\}$ forms a subtree of T.

The *width* of the tree decomposition $\mathcal{T} = (\mathcal{X}, T)$ is $\max_{i \in I} |X_i| - 1$. The *treewidth* of a graph G is the minimum width of a tree decomposition of G. We use the term vertex for the elements of V, nodes for the elements of I, and bags for the elements of $\mathcal{X}$. We say the set X_i is the corresponding bag to node $i \in I$ in tree T.

Computing the treewidth of arbitrary graphs is an NP-hard problem [1]. However, if the treewidth is bounded, it can be computed in polynomial time [7].

A tree decomposition $\mathcal{T}$ of a graph can be simply made rooted by designating a bag as the root bag. A rooted tree decomposition is called *nice* [8] whenever each bag $X_i \in \mathcal{X}$ is one of the following types:

- *leaf bag:* the node i has no child in T,
- *forget bag:* the node i has exactly one child j where $X_i \subseteq X_j$ and $|X_i| = |X_j| - 1$,

- *introduce bag:* the node i has exactly one child j such that $X_j \subseteq X_i$ and $|X_i| = |X_j| + 1$,
- *join bag:* the node i has exactly two children j and j' where $X_i = X_j = X_{j'}$.

Given a tree decomposition $\mathcal{T}$ of G of width $\mathrm{tw}(G)$ with $\mathcal{O}(n)$ bags, a nice tree decomposition of width $\mathrm{tw}(G)$ can be obtained with at most $\mathcal{O}(4n)$ bags in $\mathcal{O}(n)$ [8].

There are many decision problems on graphs that are NP-complete in general but can be solved in polynomial time for graphs with bounded treewidth using dynamic programming. See [12] for further details.

2 New result

Theorem 6 *There exists an algorithm to compute the toughness of a graph G with running time $\mathcal{O}(\mathrm{tw}(G)^{2\mathrm{tw}(G)} \cdot |V(G)|^3)$.*

SKETCH OF PROOF: First notice that for a non-complete, connected graph G on n vertices, the toughness $\tau(G)$ is a rational number $\frac{p}{q}$ with $1 \leq p, q \leq n$. So if we have an algorithm that computes the maximum number of components after the removal of s vertices for any integer $0 \leq s < n$, then the toughness can be easily determined by finding the minimum ratio.

To solve this modified problem, we use a dynamic programming approach, similar to many of the known algorithms in the literature.

Take a nice tree decomposition $\mathcal{T} = (\mathcal{X}, T)$ with $T = (I, F)$. Let V_i denote the set of all vertices of G appearing in bags that are descendants of some node i in T, and $G_i := G[V_i]$.

Compute the following information in the rooted tree T for each vertex $i \in I = V(T)$ in a bottom up order (i.e. first for the leafs, then for their parents, etc.):
$\textsc{Mnc}(i, s, Q, \mathcal{P})$: the maximum number of components of $G_i - S$ where the maximum is taken for all sets $S \subseteq V_i$ having

- $|S| = s$,
- $S \cap X_i = Q$, and
- $\mathcal{P}$ is the partition of $X_i - Q = X_i - S$ where two vertices belong to the same set if and only if they belong to the same component of the graph $G_i - S$.

For every $i \in I$, compute $\textsc{Mnc}(i, s, Q, \mathcal{P})$ for each possible value of $0 \leq s < n$, set $Q \subseteq X_i$ and partition $\mathcal{P}$ using the previously computed information for the child/children of i. The total size of information for one vertex of T is $\mathcal{O}\left(|V(G)| \cdot \mathrm{tw}(G)^{\mathrm{tw}(G)}\right)$.

Finally, for the root r of the tree, compute

$$\tau(G) = \min\left\{ \frac{s}{\textsc{Mnc}(r, s, Q, \mathcal{P})} \;\middle|\; 0 \leq s < n,\ \textsc{Mnc}(r, s, Q, \mathcal{P}) \geq 2 \right\}.$$

To complete the description of the algorithm, we need to be able to compute all of these values using the values of the children of every node in the tree decomposition. For the leaf bags, this is trivial, for the forget bags, it is straightforward. For the introduce bags, this computation is somewhat more complicated, but it is the most complex for the join bags. □

Corollary 7 *The toughness can be computed in polynomial time for graphs with bounded treewidth.*

Corollary 8 *Toughness is fixed-parameter tractable parameterized with treewidth.*

It is an open question weather there exists a faster algorithm with time bound $\mathcal{O}\left(c^{\mathcal{O}(\mathrm{tw}(G))} \cdot |V(G)|^3\right)$ for some constant c to compute toughness G, as do such algorithms for many similar problems in the literature. It seems that the known techniques do not work for toughness. For some other similar problems, it is proven that no such algorithm exists under the Exponential Time Hypothesis. Unfortunately, these techniques do not seem to work either. So finding a better algorithm or proving that such an algorithm does not exist seems to be a challenging open problem.

References

[1] S. Arnborg, D.G. Corneil, A. Proskurowski, Complexity of finding embeddings in a k-tree, *SIAM Journal on Algebraic Discrete Methods* **8** (2):277–284 (1987)

[2] D. Bauer, S.L. Hakimi, E. Schmeichel, Recognizing tough graphs is NP-hard, *Discrete Applied Mathematics* **28**:191–195 (1990)

[3] D. Bauer, A. Morgana, E. Schmeichel, On the complexity of recognizing tough graphs, *Discrete Mathematics* **124**:13–17 (1994)

[4] D. Bauer, J. van den Heuvel, A. Morgana, E. Schmeichel, The complexity of recognizing tough cubic graphs, *Discrete Applied Mathematics* **79**:35–44 (1997)

[5] D. Bauer, J. van den Heuvel, A. Morgana, E. Schmeichel, The complexity of toughness in regular graphs, *Congressus Numerantium* **130**:47–61, (1998)

[6] H.L. Bodlaender, A tourist guide through treewidth, *Acta Cybernetica* **11** (1-2):1–21 (1993)

[7] H.L. Bodlaender, A linear time algorithm for finding tree-decompositions of small treewidth, *SIAM Journal on Computing* **25** (6):1305–1317 (1996)

[8] H.L. Bodlaender, P. Bonsma, D. Lokshtanov, The fine details of fast dynamic programming over tree decompositions, *In:* International Symposium on Parameterized and Exact Computation, Springer, 41–53 (2013)

[9] H. Broersma, How tough is toughness?, *Bulletin of EATCS* **117**:28–52 (2015)

[10] H.J. Broersma, V. Patel, A. Pyatkin, On toughness and hamiltonicity of $2K_2$-free graphs, *Journal of Graph Theory* **75**(3):244–255 (2014)

[11] V. Chvátal, Tough graphs and hamiltonian circuits, *Discrete Mathematics* **5**(3):215–228 (1973)

[12] M. Cygan, F.V. Fomin, Ł. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, S. Saurabh, Parameterized Algorithms, *Springer*, ISBN 978-3-319-21274-6 (2015)

[13] B. Jackson, P. Katerinis, A characterization of 3/2-tough cubic graphs, *Ars Combinatoria* **38**:145–148 (1994)

[14] G.Y. Katona, K. Varga, Strengthening Some Complexity Results on Toughness of Graphs, *Discussiones Mathematicae Graph Theory* **43**(2):401–419 (2023)

[15] D. Kratsch, T. Kloks, H. Müller, Computing the toughness and the scattering number for interval and other graphs, *INRIA Research Report* RR–2237 (1994)

[16] D. Kratsch, J. Lehel, H. Müller, Toughness, hamiltonicity and split graphs, *Discrete Mathematics* **150**(1–3):231–245 (1996)

[17] M.M. Matthews, D.P. Sumner, Hamiltonian results in $K_{1,3}$-free graphs, *Journal of Graph Theory* **8**(1):139–146 (1984)

[18] N. Robertson, P.D. Seymour, Graph minors. I. Excluding a forest, *Journal of Combinatorial Theory Series B* **35**(1):39–61 (1983)

[19] N. Robertson, P.D. Seymour, Graph minors. II. Algorithmic aspects of tree-width, *Journal of Algorithms* **7**(3):309–322 (1986)

[20] G.J. Woeginger, The toughness of split graphs, *Discrete Mathematics* **190**(1–3):295–297 (1998)

On the size of highly redundantly rigid graphs

Csaba Király

Department of Operations Research
ELTE Eötvös Loránd University
and
ELKH-ELTE Egerváry Research Group on
Combinatorial Optimization
Eötvös Loránd Research Network (ELKH)
Pázmány Péter sétány 1/C, Budapest, 1117,
Hungary
csaba.kiraly@ttk.elte.hu

Abstract: A graph is called k-vertex-redundantly rigid in $\mathbb{R}^d$ if it remains rigid in $\mathbb{R}^d$ even after the deletion of any set of at most $k-1$ vertices. k-edge-redundant rigidity can be defined similarly. The minimum degree in a k-vertex/edge-redundantly rigid graph in $\mathbb{R}^d$ is at least $k+d-1$ which implies a lower bound on the number of edges in a k-vertex-redundantly rigid graph on n vertices, in terms of k, d, and n. For sufficiently large k, we show the tightness of this bound for all d and infinitely many values of n. Our results have broader applications, including providing tight lower bounds for other problems such as the size of k-edge-redundantly rigid, or k-vertex/edge-redundantly globally rigid graphs. We also provide almost tight upper bound for the size of minimally k-edge-redundantly rigid graphs in $\mathbb{R}^d$ on n vertices for all possible values of k, d, and n, by extending an idea applied earlier for k-vertex-redundant rigidity.

Keywords: rigidity, vertex-redundant rigidity, global rigidity

1 Introduction

A **d-dimensional framework** is a pair (G,p), where $G=(V,E)$ is a graph and p is a map from V to $\mathbb{R}^d$. We will also refer to (G,p) as a **realization** of G. Two realizations (G,p) and (G,q) of G are **equivalent** if $||p(u)-p(v)||=||q(u)-q(v)||$ holds for all pairs u,v with $uv\in E$. Frameworks (G,p) and (G,q) are **congruent** if $||p(u)-p(v)||=||q(u)-q(v)||$ holds for all pairs u,v with $u,v\in V$. We say that (G,p) is **globally rigid** in $\mathbb{R}^d$ if every d-dimensional framework which is equivalent to (G,p) is also congruent to (G,p). A framework (G,p) is **rigid** if there exists an $\varepsilon>0$ such that, if (G,q) is equivalent to (G,p) and $||p(u)-q(u)||<\varepsilon$ for all $u\in V$, then (G,q) is congruent to (G,p).

We assign to (G,p) a matrix, called the **rigidity matrix $\boldsymbol{R(G,p)}\in\mathbb{R}^{|E|\times d|V|}$** that is defined as follows. We assign a row of $R(G,p)$ to each edge $uv\in E$ and d columns to each $v\in V$. Let the d entries of $R(G,p)$ in the row assigned to $uv\in E$ and d colums assigned to $w\in V$ be 0_d if $w\neq u,v$, $p(u)-p(v)$ if $w=u$, and $p(v)-p(u)$ if $w=v$. (G,p) is called **infinitesimally rigid** if $\mathrm{rank}(R(G,p))=d|V|-\binom{d+1}{2}$. Infinitesimal rigidity of (G,p) implies its rigidity but the other direction is not always true. The realization p is called **generic** if the elements of the set $\{p(v_i)_j : i=1,...,|V|,\ j=1,...,d\}$ are algebraically independent over $\mathbb{Q}$. It is known that for generic realizations of a graph rigidity and infinitesimal rigidity are equivalent.

A graph G is called **rigid** in $\mathbb{R}^d$ if it has an infinitesimally rigid realization in $\mathbb{R}^d$. If p_0 is a generic realization of G, then it follows by the definition of genericity that $\mathrm{rank}(R(G,p_0))=\max\{\mathrm{rank}(R(G,p)) : p:V\to\mathbb{R}^d\}$. Hence generic realizations of a rigid graph are always infinitesimally rigid. A graph G is called **globally rigid** in $\mathbb{R}^d$ if it has a generic globally rigid realization in $\mathbb{R}^d$. By the results of Connelly

[3] Gortler, Healy and Thurston [5], if a graph has a generic globally rigid realization, then all of its generic realizations are globally rigid. We refer for more details to [6, 23]. As parallel edges and loops are meaningless in (the above version of) rigidity theory, we assume throughout this paper that all graphs are *simple*, that is, contain no parallel edges or loops.

We call a graph $G = (V, E)$ **k-vertex-redundantly rigid in $\mathbb{R}^d$** (or shortly **$[k, d]$-rigid**) if $G - U$ is rigid in $\mathbb{R}^d$ for every $U \subset V$ of cardinality at most $k - 1$. It is well-known (see [10] for a proof) that a graph G on at least $k + 1$ vertices is $[k, d]$-rigid if and only if $G - U$ is rigid in $\mathbb{R}^d$ for every $U \subset V$ of cardinality exactly $k - 1$. We say that a $[k, d]$-rigid graph $G = (V, E)$ is **minimally $[k, d]$-rigid** if $G - e$ is not $[k, d]$-rigid for each $e \in E$. The main focus of this paper is on the edge number (that is, the size) of minimally $[k, d]$-rigid graphs. When $k = 1$, it is well-known that all minimally $[1, d]$-rigid graphs on n vertices (of cardinality at least $d + 1$) have the same size: $(d + 1)n - \binom{d+1}{2}$ (see Whiteley [23]). However, as it was shown in [15], no such fix number exist for larger values of k for any pair of d and $n \geq k + d$. Hence the goal is to give (tight) lower and upper bounds to the size of $[k, d]$-rigid graphs on n vertices. The first tight lower bound was given by Servatius [20] for $k = d = 2$. Motevallian, Yu and Anderson [19] (see also [13]) gave tight lower bound for $[k, d] = [3, 2]$. In [15] general lower bounds were given for all pairs of $[k, d]$ and their tightness was proven for $k = 2$ for all d, and for $k = d = 3$. It is obvious that the minimum degree in graph which is rigid in $\mathbb{R}^d$ is at least d when it has at least $d + 1$ vertices. This implies that the minimum degree in a $[k, d]$-rigid graph (on at least $k + d$ vertices) must be at least $k + d - 1$ and hence we get the following lower bound to the size of such graphs.

Theorem 1 *[15] Let $G = (V, E)$ be a $[k, d]$-rigid graph on at least $d + k$ vertices. Then*

$$|E| \geq \left\lceil \frac{k + d - 1}{2} |V| \right\rceil. \quad \square$$

Since a rigid graph in $\mathbb{R}^d$ on n vertices induces at least $dn - \binom{d+1}{2}$ edges, the above lower bound cannot be tight when $k \leq d$. Furthermore, [15, Theorem 5] implies that it is not tight for $k = d + 1$ when $d \geq 2$. However, it was conjectured in [15] that the bound is tight whenever $k \geq d + 2$. Jordán [10] verified this conjecture for $d = 2$, and Jordán, Poston and Roach [14] proved it for $d = 3$. In this note, we verify the conjecture for all $d \geq 4$ for the cases where $k \geq \frac{d^2}{4} + 2d + 2$.

Similarly to k-vertex redundant rigidity we can define **k-edge-redundant rigidity in $\mathbb{R}^d$** (or shortly **$[k, d]$-edge rigidity**), and **k-edge/vertex-redundant global rigidity in $\mathbb{R}^d$** (or shortly **$[k, d]$-edge/vertex global rigidity**). For example, a graph $G = (V, E)$ is called $[k, d]$-edge globally rigid if $G - F$ is rigid for all $F \subseteq E$ of cardinality (at most) $k - 1$. In [10, 14], tight lower bounds were given for the edge number of all of these type of graphs for $d = 2, 3$ for (almost) all k (see Table 1 in Section 5 for the missing cases). As for $[k, d]$-rigidity, we can obtain a trivial bound based on the minimum degree in rigid and globally rigid graphs. We have used before that the minimum degree in a rigid graph in $\mathbb{R}^d$ on at least $d + 1$ vertices is at least d. On the other hand, by a result of Hendrickson [7], a globally rigid graph in $\mathbb{R}^d$ on at least $d + 2$ vertices must be $d + 1$-connected, and hence the minimum degree of such a graph must be at least $d + 1$. These imply the following.

Theorem 2 *[10, 16] Let $G = (V, E)$ be a $[k, d]$-edge rigid graph on at least $d+1$ vertices. Let $G' = (V', E')$ be a $[k, d]$-edge globally rigid graph on at least $d + 1$ vertices. Let $G'' = (V'', E'')$ be a $[k, d]$-vertex globally rigid graph on at least $d + k$ vertices. Then*

$$|E| \geq \left\lceil \frac{k + d - 1}{2} |V| \right\rceil, \ |E'| \geq \left\lceil \frac{k + d}{2} |V'| \right\rceil, \ \textit{and} \ |E''| \geq \left\lceil \frac{k + d}{2} |V''| \right\rceil. \quad \square$$

As an application of our main result on the tightness of the bound given in Theorem 1, we show in the end of Section 3 that the bounds given in Theorem 2 are also tight if k is sufficiently large in term of d.

Besides seeking for lower bounds, it is natural to ask how large the size of a minimally $[k, d]$-rigid ($[k, d]$-edge rigid, $[k, d]$-edge/vertex globally rigid, respectively) graph on n vertices can be. In [15], a tight upper bound was given to the size of $[k, d]$-rigid graphs for all pairs of k and d, as follows.

Theorem 3 *[15] Let $G = (V, E)$ be a minimally $[k, d]$-rigid graph on at least $d + k$ vertices. Then*

$$|E| \leq (d + k - 1)|V| - \binom{d+k}{2}.$$

Moreover, this bound is tight, when $d \geq 2$.

We note that for $d = 1$ rigidity is equivalent to connectivity and hence $[k, 1]$-rigidity is equivalent to k-connectivity. In this case, the tight upper bound can be obtained by a result of Mader [18].

Contrary to vertex-redundant rigidity, just a few partial results exists on the three other variants of the problem. Jordán [8, 11] gave upper bound for the size of minimally $[2, 2]$-edge rigid simple graphs on vertex set V: the upper bound is $3|V| - 9$ for $|V| \geq 7$ which is tight as the complete bipartite graphs $K_{3,n-3}$ show. For larger values of k and d, no tight upper bound is known for the size of minimally $[k, d]$-edge rigid graphs. Jordán [11] proved that $2d|V|$ minus a constant depending on d is an upper bound for this size when $k = 2$. However, this bound is far from the conjectured tight upper bound which is $(d + 1)|V|$ minus a constant depending on d (see also [4]). In Section 4, by using the proof technique of Theorem 3 in [15], we show that the same upper bound as in Theorem 3 holds for the size minimally $[k, d]$-edge rigid simple graphs. This statement verifies (and extends for higher values of k) the above conjecture. f

For minimal $[k, d]$-vertex global rigidity, even the upper bound for $k = 1$ and $d \geq 3$ had been open [9, 16] until recently Garamvölgyi and Jordán [4] proved that the tight upper bound for the edge number of a minimally $[1, d]$-vertex globally rigid graph on at least $d + 2$ vertices is $(d + 1)|V| - \binom{d+2}{2}$, moreover, this bound is only tight for the complete graph K_{d+2}. (Although, for larger values of $|V|$, this bound is almost tight which follows by the global rigidity of $K_{d+1,n-d-1}$ for $n \geq \binom{d+2}{2}$ [12].) For higher values of k, the following result along with the proof techniques of [15] provides almost tight upper bound for the size of minimally $[k, d]$-vertex globally rigid graphs when $d = 1, 2$. We say that an edge is an **$\mathcal{R}_d$-bridge** (of G) if its corresponding row in the rigidity matrix $R(G, p)$ of a generic realization of G in $\mathbb{R}^d$ is linearly independent from the set of the other rows of the matrix.

Theorem 4 *[4] Let G be a graph which is globally rigid in $\mathbb{R}^d$ where $d = 1, 2$. Assume that $G - e$ is not globally rigid for an edge e. Then e is an $\mathcal{R}_{d+1}$-bridge of G.*

It was conjectured in [9, 16] that this statement is also true for higher values of d. The upper bound which arises by using this result is the following.

Theorem 5 *[4] Let $G = (V, E)$ be a minimally $[k, d]$-vertex globally rigid simple graph on at least $d+k+1$ vertices where $d = 1$ or 2. Then*

$$|E| \leq (d + k)|V| - \binom{d+k+1}{2}. \quad \square$$

The complete graph K_{d+k+1} shows that the above bound is tight for $|V| = d+k+1$, and the complete bipratite graphs $K_{d+k,n-d-k}$ ($n \geq \binom{d+k+1}{2}$ show that the bound is almost tight for higher values of $|V|$. In Section 4, we show that the same upper bound holds for the size of minimally $[k, d]$-edge globally rigid simple graphs if $d = 1$ or 2. Again, the bound is almost tight. Finally, in Section 5, we give a brief overview of the corresponding results, summarize the open problems of the topic.

2 The vertex splitting operation

Let $G = (V, E)$ be a graph, let $v_1 \in V$, let $v_1v_2, ..., v_1v_d$ be $d - 1$ designated edges incident with v_1, and let E_0 and E_1 be a bipartition of the remaining edges incident with v_1. The **(d-dimensional) vertex splitting** operation at v_1 removes the edges in E_0, adds a new vertex v_0, and adds a new set of edges $\{v_0v_1, v_0v_2, ..., v_0v_d\} \cup \{v_0v : v_1v \in E_0\}$ to G. The special cases of vertex splitting where $E_0 = \emptyset$ or $|E_0| = 1$ are called **0-extension** and **1-extension**, respectively. Whiteley [22] proved that the vertex splitting operation preserves the rigidity of graphs in $\mathbb{R}^d$.

Theorem 6 ([22]) *Let G be a rigid graph in $\mathbb{R}^d$ and let G' be the result from applying a d-dimensional vertex splitting to G. Then G' is rigid in $\mathbb{R}^d$.* □

Next, let G be a graph, let $v_1 \in V$, let $v_1v_2, ..., v_1v_{d+1}$ be d designated edges incident with v_1, and let E_0 and E_1 be a bipartition of the remaining edges incident with v_1. The **(d-dimensional) extended vertex split** at v_1 removes the edges in E_0, adds a new vertex v_0, and adds the new edges in $E_2 = \{v_0v_2, ..., v_0v_{d+1}\} \cup \{v_0v : v_1v \in E_0\}$ to G. Berenchtein, Chavez and Whiteley [1] proved that the 2-dimensional extended vertex split preserves the rigidity of graphs in $\mathbb{R}^2$. Whiteley [22] also noted (without proof) that it preserves rigidity in $\mathbb{R}^3$. Here we extend these results by showing that the d-dimensional extended vertex split preserves the rigidity in $\mathbb{R}^d$.

Theorem 7 *Let $G = (V, E)$ be a rigid graph in $\mathbb{R}^d$ and let $G' = (V + v_0, E')$ be the result of a d-dimensional extended vertex split applied to G at vertex v_1. Then G' is rigid in $\mathbb{R}^d$.*

Proof: Let p be a generic realization of G and let us define $p(v_0) := p(v_1)$. We will show that (G', p) is infinitesimally rigid, that is, $\text{rank}(R(G', p)) = d(|V| + 1) - \binom{d+1}{2} = \text{rank}(R(G, p)) + d$.

Let $G'' = (V + v_0, E'')$ be the result of a d-dimensional extended vertex split applied to G at vertex v_1 with $E_0 = \emptyset$. Observe that

$$R(G'', p) = \begin{pmatrix} p(v_0) - p(v_2) & * \\ \cdots & \cdots \\ p(v_0) - p(v_{d+1}) & * \\ 0 & R(G, p) \end{pmatrix}.$$

Note also that it follows from the genericity of $p|_V$ and $p(v_0) = p(v_1)$ that $p(v_0)-p(v_2), \ldots, p(v_0)-p(v_{d+1})$ form a basis of $\mathbb{R}^d$. These imply that $\text{rank}(R(G'', p)) = \text{rank}(R(G, p)) + d$.

Next, observe that $R(G'', p)$ and $R(G', p)$ differ at the rows corresponding to edges in E_0 since the copy of these edges in G' ends at v_0. In fact, their difference is that the row corresponding to such an edge ending at v has $p(v_0) - p(v) = p(v_1) - p(v)$ in the first d columns (corresponding to v_0) in $R(G', p)$ and zeros in the next d columns corresponding to v_1 while, in $R(G'', p)$, the first d coordinates are zeros and the second d consists of the vector $p(v_1) - p(v)$, all the other entries in these rows are equal to each other. To show that the rank of the two matrices are equal, it is enough to show the following claim.

Claim 8 *Let $q \in \mathbb{R}^d$. Then the vector $(q, -q, 0_{d(|V|-1)})$ is in the row space of both $R(G', p)$ and $R(G'', p)$.*

Proof: Both matrices contain the rows $q_i = (p(v_0) - p(v_i), 0, \ldots, 0, p(v_i) - p(v_0), 0, \ldots 0)$ and $q'_i = (0_d, p(v_0) - p(v_i), 0, \ldots, 0, p(v_i) - p(v_0), 0, \ldots 0)$ for each $i \in \{2, \ldots, d+1\}$ since v_0v_i and v_1v_i are edges of both of G' and G'' and $p(v_0) = p(v_1)$. As we have seen before, $\{p(v_0) - p(v_i) : i \in \{2, \ldots, d+1\}\}$ is a basis of $\mathbb{R}^d$. Hence there exist constants λ_i for $i \in \{2, \ldots, d+1\}$ such that $\sum_{i=2}^{d+1} \lambda_i(p(v_0) - p(v_i)) = q$. Now, $\sum_{i=2}^{d+1} \lambda_i q_i - \lambda_i q'_i = (q, -q, 0_{d(|V|-1)})$, which proves the claim. □

Let $v_1v \in E_0$. The difference of the rows of $R(G'', p)$ and $R(G', p)$ corresponding to v_1v and v_0v, respectively, is $(p(v_0) - p(v), p(v) - p(v_0), 0_{d(|V|-1)})$ which is in the row space of both matrices by the above claim. This implies that the row space of the two matrices coincide and hence they have the same rank. This finishes our proof. □

3 $[k, d]$-rigid graphs with minimum size

We shall also use the following result on the rigidity of complete bipartite graphs in our construction.

Theorem 9 (Whiteley [21] and [23, Theorem 11.2.1]) *Let $m, n \geq 2$ be two integers. The complete bipartite graph $K_{m,n}$ is $[1, d]$-rigid if and only if $m + n \geq \binom{d+2}{2}$ and $m, n \geq d + 1$.* □

In fact, the results of Whiteley [21] can also be applied for the case where we add extra edges to a complete bipartite graph. In this case the lower bound $\binom{d+2}{2}$ on the number of vertices can be reduced by the number of new edges. Since this is only proved exactly for the case where $d = 3$ in [21], we give a direct proof for the following (simpler) statement that we shall use in our construction.

Lemma 10 *Let $G = (S_1 \cup S_2 \cup T_1 \cup T_2, E)$ be the union of the complete bipartite graph $K_{S_1 \cup S_2, T_1 \cup T_2}$ and of the complete graph $K_{S_1 \cup T_1}$. Assume that $\ell := |S_1 \cup T_1| \geq d$, and $|S_2| = |T_2| \geq d + 1 + \lfloor \frac{(d-\ell/2)^2 - \ell}{2} \rfloor$. Then G is $[1, d]$-rigid.* □

Proof: We shall use 0- and 1-extensions in the proof. Let us start with $K_{S_1 \cup T_1}$, which is rigid as it is complete. (Note that if $|S_1| \geq d$ or $|T_1| \geq d$, then a rigid subgraph of G can be got from this complete graph by using 0-extensions. Hence, from now on, we will assume that $|S_1|, |T_1| < d$.) Next we add subsequently $d - |S_1|$ vertices from S_2 and $d - |T_1|$ vertices from T_1 by using 0-extensions by adding as much edges of G as we can and some extra, **auxiliary** edges. Note that if we add the vertices from S_2 first, then we need to add $(d - |S_1|)(d - |T_1|)$ auxiliary edges to the side of S_1 during the 0-extensions and 0 to the other side. (These edges will be removed later by using 1-extensions.) Suppose that in a sequence of vertices (of the above 0-extensions), there is a S_2 vertex followed by a T_2 vertex. If we change the order by swapping these two vertices, the number of auxiliary edges added to the side of S_1 decreases by one, while the number of auxiliary edges added on the other side increases by one. This way we can assume that we add $\lfloor (d - |S_1|)(d - |T_1|)/2 - (|S_1| - |T_1|)/2 \rfloor$ auxiliary edges to the side of S_1 and $\lceil (d - |S_1|)(d - |T_1|)/2 + (|S_1| - |T_1|)/2 \rceil$ edges to the other side. Before starting 1-extensions, we need to add one more vertex from T_2 by using a 0-extension (without adding any new auxiliary edges). Next we add $\lceil (d - |S_1|)(d - |T_1|)/2 + (|S_1| - |T_1|)/2 \rceil$ vertices from S_2 by using 1-extensions and deleting the auxiliary edges from the side of T_1, and after this we add $\lfloor (d - |S_1|)(d - |T_1|)/2 - (|S_1| - |T_1|)/2 \rfloor$ vertices from T_2 by using 1-extensions and deleting the auxiliary edges from the side of S_1. Finally, we add the rest of vertices (if any) by using 0-extensions. This way we get a rigid subgraph of G. Note that we needed to add at least $d - |S_1| + \lceil (d - |S_1|)(d - |T_1|)/2 + (|S_1| - |T_1|)/2 \rceil$ vertices from S_2 and $d - |T_1| + 1 + \lfloor (d - |S_1|)(d - |T_1|)/2 - (|S_1| - |T_1|)/2 \rfloor$ vertices from T_2 which is possible by our assumption on the cardinality of S_2 and T_2. □

The construction when $k+d$ is odd. Let us take n sets $V_1, \ldots, V_n$ of cardinality $\frac{k+d-1}{2}$ and let E_i be the edge set of the complete bipartite graph with color classes V_i and V_{i+1} for $i = 1, \ldots n$, where we use the notation $V_{n+i} := V_i$ $(i = 1, \ldots, n)$. Let $\boldsymbol{K_{\frac{k+d-1}{2} \times n}}$ denote the graph on vertex set $V = V_1 \cup \ldots, V_n$ with edge set $E = E_1 \cup \ldots, E_n$. Our main result is the following.

Theorem 11 *Assume that $d \geq 2$, $n \geq 5d + 1$, $k \geq \frac{d^2}{4} + 2d - 1$, and $k + d$ is odd. Then $K_{\frac{k+d-1}{2} \times n}$ is $[k, d]$-rigid.*

Since $K_{\frac{k+d-1}{2} \times n}$ is $(k + d - 1)$-regular, its $[k, d]$-rigidity implies that it is extremal for the bound of Theorem 1 on the edge number of $[k, d]$-rigid graphs.

Proof: Let $S \subset V$ be a subset of vertices with $|S| = k - 1$. We need to show that $K_{\frac{k+d-1}{2} \times n} - S$ is rigid in $\mathbb{R}^d$.

Note that $k \geq \frac{d^2}{4} + 2d - 1$, $d \geq 2$, and the fact that $k + d$ is odd imply that $k \geq 3d - 1$ also holds. For each triple $i_1, i_2, i_3 \in \{1, \ldots, n\}$, $|V_{i_1} \cup V_{i_2} \cup V_{i_3} - S| \geq 3\frac{k+d-1}{2} - (k - 1) = \frac{k+3d-1}{2} \geq 3(d - 1) + 1$ holds by $k \geq 3d - 1$. This implies that $|V_i - S| \geq d$ holds for all but at most two $i \in \{1, \ldots, n\}$. Similarly, for each $i_1, i_2, i_3, i_4, i_5, i_6 \in \{1, \ldots, n\}$, $|V_{i_1} \cup V_{i_2} \cup V_{i_3} \cup V_{i_4} \cup V_{i_5} \cup V_{i_6} - S| \geq 6\frac{k+d-1}{2} - (k - 1) = \frac{4k+6d-4}{2} \geq 9d - 4 \geq 6d + 1$ holds by $k \geq 3d - 1$ and $d \geq 2$. This implies that $|V_i - S| \geq d + 1$ holds for all but at most five $i \in \{1, \ldots, n\}$. Thus, by the pigeonhole principle and $n \geq 5d + 1$, there exists an index $i_0 \in \{1, \ldots, n\}$ such that $|V_{i_0+j} - S| \geq d + 1$ holds for $j = 0, \ldots, d - 1$. By relabeling the sets V_i's cyclically, we can assume that $i_0 = 1$.

We first show that $G = (K_{\frac{k+d-1}{2}\times n} - S)[V_1 \cup \cdots \cup V_d]$ is rigid in $\mathbb{R}^d$. It is enough to show the rigidity of an induced subgraph G' of G for which $|V(G') \cap (V_i - S)| = d+1$ holds for each $i \in \{1, \ldots, d\}$, since G arises from this subgraph by 0-extensions and edge additions.

Claim 12 *G' is rigid.*

PROOF: Let $V(G') \cap (V_i - S) = \{v_1^i, \ldots, v_{d+1}^i\}$ for $i = 1, \ldots, d$. Observe that if we contract all of the sets $\{v_1^2, v_1^4, \ldots, v_1^{2\lfloor \frac{d}{2} \rfloor}\}, \ldots, \{v_{d+1}^2, v_{d+1}^4, \ldots, v_{d+1}^{2\lfloor \frac{d}{2} \rfloor}\}$ into single vertices, then the arising graph is the complete bipartite graph $K_{d+1,\lceil \frac{d}{2} \rceil (d+1)}$ which is rigid by Theorem 9. On the other hand G' arises from this graph by using extended vertex splitting and edge additions and hence G' is rigid by Theorem 7. □

Next, if $|V_i - S| \geq d$ holds for all but one $i \in \{1, \ldots, n\}$, then we can add all the vertices of $K_{\frac{k+d-1}{2}\times n} - S$ by 0-extensions to G' in a row to prove its rigidity. Furthermore, if we try to use 0-extensions in the case where there are two small sets, we only get stuck if there exist two indices $d+1 \leq i_1 < i_2 \leq n$ with $i_2 - i_1 \geq 3$ such that $|V_{i_1} - S| \leq d-1$ and $|V_{i_2} - S| \leq d-1$. (Note that, when $i_2 - i_1 = 2$, the vertices in V_{i_1+1} have at least d neighbors in $V_{i_1} \cup V_{i_2} - S$.) In this remaining case, like before, we can use the rigidity of G' and 0-extensions to prove the rigidity of $K_{\frac{k+d-1}{2}\times n}[V_1 \cup \cdots \cup V_d \cup V_{d+1} \cup \cdots \cup V_{i_1} \cup V_{i_2} \cup \cdots \cup V_n - S]$. This implies that we may add the edge set F of the complete graph $K_{V_1 \cup \cdots \cup V_d \cup V_{d+1} \cup \cdots \cup V_{i_1} \cup V_{i_2} \cup \cdots \cup V_n - S}$ without increasing the rank of the corresponding rigidity matrix (since the part corresponding to the vertices in $V_1 \cup \cdots \cup V_{i_1} \cup V_{i_2} \cup \cdots \cup V_n - S$ has already maximum rank), that is, $K_{\frac{k+d-1}{2}\times n} - S$ is rigid if and only if $(K_{\frac{k+d-1}{2}\times n} - S) + F$ is rigid. Next we prove the rigidity of $(K_{\frac{k+d-1}{2}\times n} - S + F)[V_{i_1} \cup \cdots \cup V_{i_2}]$.

Now $|V_{i_1} - S| \leq d-1$ and $|V_{i_2} - S| \leq d-1$ imply that $|V_{i_1} \cap S| \geq \frac{k+d-1}{2} - (d-1) = \frac{k-d+1}{2}$ and $|V_{i_2} - S| \geq \frac{k-d+1}{2}$. Hence $|S \cap V_i| \leq d-2$ and hence $|V_i - S| \geq \frac{k-d+3}{2}$ holds for each index $i_1, i_2 \neq i \in \{1, \ldots, n\}$. Like in the previous case, for each $i_1 < i < i_2$, we denote the first $\frac{k-d+3}{2}$ elements of $V_i - S$ by $\{v_1^i, \ldots, v_{\frac{k-d+3}{2}}^i\}$ and delete the rest of its elements. Next we contract the sets $\{v_1^{i_1+1}, v_1^{i_1+3}, \ldots, v_1^{i_1+2\lfloor \frac{i_2-i_1}{2} \rfloor - 1}\}, \ldots, \{v_{\frac{k-d+3}{2}}^{i_1+1}, v_{\frac{k-d+3}{2}}^{i_1+3}, \ldots, v_{\frac{k-d+3}{2}}^{i_1+2\lfloor \frac{i_2-i_1}{2} \rfloor - 1}\}$ and $\{v_1^{i_1+2}, v_1^{i_1+4}, \ldots, v_1^{i_1+2\lceil \frac{i_2-i_1}{2} \rceil - 2}\}, \ldots, \{v_{\frac{k-d+3}{2}}^{i_1+2}, v_{\frac{k-d+3}{2}}^{i_1+4}, \ldots, v_{\frac{k-d+3}{2}}^{i_1+2\lceil \frac{i_2-i_1}{2} \rceil - 2}\}$ into single vertices $u_1^1, \ldots, u_{\frac{k-d+3}{2}}^1$ and $u_1^2, \ldots, u_{\frac{k-d+3}{2}}^2$. (Recall that we are assuming now that $i_2 - i_1 \geq 3$ and hence we indeed get at least $2\frac{k-d+3}{2}$ contracted vertices.)

The resulting graph after the contraction is the union of the complete graph $K_{(V_{i_1} \cup V_{i_2}) - S}$ and either the complete bipartite graph $K_{\frac{k-d+3}{2}+|V_{i_1}-S|, \frac{k-d+3}{2}+|V_{i_2}-S|}$ or $K_{\frac{k-d+3}{2}+|V_{i_1}-S|+|V_{i_2}-S|, \frac{k-d+3}{2}}$. Note that $|V_{i_1} \cup V_{i_2} - S| \geq k+d-1-(k-1) \geq d$. Hence the contracted graph is rigid by Theorem 10. This also implies the rigidity of $(K_{\frac{k+d-1}{2}\times n} - S + F)[V_{i_1} \cup \cdots \cup V_{i_2}]$ by using Theorem 7 (and 0-extensions). The rest of the vertices of $K_{\frac{k+d-1}{2}\times n} - S + F$ Hence $K_{\frac{k+d-1}{2}\times n} - S + F$ is indeed rigid which, as we have seen earlier, implies the rigidity of $K_{\frac{k+d-1}{2}\times n} - S$, finishing our proof. □

The construction when $k+d$ is even. Let $\boldsymbol{K'_{\frac{k+d-2}{2}\times 2n}}$ be obtained from $K_{\frac{k+d-2}{2}\times 2n}$ by adding a matching between V_i and V_{i+n} (called long diagonals) for each $i = 1, \ldots n$. (Again, we use the notation $V_{2n+i} := V_i$ $(i = 1, \ldots, 2n)$.) We claim now the following.

Theorem 13 *Assume that $d \geq 2$, $2n \geq 5d+1$, $k \geq \frac{d^2}{4} + 2d + 2$, and $k+d$ is even. Then $K'_{\frac{k+d-2}{2}\times 2n}$ is $[k,d]$-rigid.*

Since $K'_{\frac{k+d-2}{2}\times 2n}$ is $(k+d-1)$-regular, its $[k,d]$-rigidity implies that it is extremal for the bound of Theorem 1 on the edge number of $[k,d]$-rigid graphs.

PROOF: Let $S \subset V$ be a subset of vertices with $|S| = k-1$ We need to show that $K'_{\frac{k+d-2}{2}\times 2n} - S$ is rigid in $\mathbb{R}^d$.

Note that $k \geq \frac{d^2}{4}+2d+2$ and $d \geq 2$ imply that $k \geq 3d+1$ holds. For each triple $i_1, i_2, i_3 \in \{1, \ldots, n\}$, $|V_{i_1} \cup V_{i_2} \cup V_{i_3} - S| \geq 3\frac{k+d-2}{2} - (k-1) = \frac{k+3d-4}{2} > 3(d-1)$ holds by $k \geq 3d+1$. This implies that $|V_i - S| \geq d$ holds for all but at most two $i \in \{1, \ldots, 2n\}$. Similarly, for each $i_1, i_2, i_3, i_4, i_5, i_6 \in \{1, \ldots, 2n\}$, $|V_{i_1} \cup V_{i_2} \cup V_{i_3} \cup V_{i_4} \cup V_{i_5} \cup V_{i_6} - S| \geq 6\frac{k+d-2}{2} - (k-1) = \frac{4k+6d-10}{2} \geq 9d-3 \geq 6d+1$ holds by $k \geq 3d+1$ and $d \geq 2$. This implies that $|V_i - S| \geq d+1$ holds for all but at most five $i \in \{1, \ldots, 2n\}$. Thus, by the pigeonhole principle and $2n \geq 5d+1$, there exists an index $i_0 \in \{1, \ldots, 2n\}$ such that $|V_{i_0+j} - S| \geq d+1$ holds for $j = 0, \ldots, d-1$. By relabeling the sets V_i's cyclically, we can assume that $i_0 = 1$.

We take an induced subgraph G' of $K'_{\frac{k+d-2}{2} \times 2n}$—like in the proof of Theorem 11—for which $|V(G') \cap (V_i - S)| = d+1$ holds for each $i \in \{1, \ldots, d\}$. G' is rigid by Claim 12. If $|V_i - S| \geq d$ holds for all but one $i \in \{1, \ldots, n\}$, then we can add all the vertices of $K'_{\frac{k+d-2}{2} \times 2n} - S$ by 0-extensions to G' in a row to prove its rigidity. Furthermore, if we try to use 0-extensions in the case where there are two small sets, we only get stuck if there exist two indices $d+1 \leq i_1 < i_2 \leq n$ with $i_2 - i_1 \geq 3$ such that $|V_{i_1} - S| \leq d-1$ and $|V_{i_2} - S| \leq d-1$. (Note that, when $i_2 - i_1 = 2$, either the vertices in V_{i_1+1} have at least d neighbors in $V_{i_1} \cup V_{i_2} - S$, or $S \subset V_{i_1} \cup V_{i_2}$, $|V_{i_1} \cup V_{i_2} - S| = d-1$ and the vertices in V_{i_1+1} have exactly d neighbors in $V_{i_1} \cup V_{i_2} \cup V_{i_1+1+n} - S$ since their neighbors along the long diagonals are not deleted.) In this remaining case, like before, we can use the rigidity of G' and 0-extensions to prove the rigidity of $K'_{\frac{k+d-2}{2} \times 2n}[V_1 \cup \cdots \cup V_d \cup V_{d+1} \cup \cdots \cup V_{i_1} \cup V_{i_2} \cup \cdots \cup V_{2n} - S]$. This implies that we may add the edge set F of the complete graph $K_{V_1 \cup \cdots \cup V_d \cup V_{d+1} \cup \cdots \cup V_{i_1} \cup V_{i_2} \cup \cdots \cup V_{2n} - S}$ without increasing the rank of the corresponding rigidity matrix (since the part corresponding to the vertices in $V_1 \cup \cdots \cup V_{i_1} \cup V_{i_2} \cup \cdots \cup V_{2n} - S$ has already maximum rank), that is, $K'_{\frac{k+d-2}{2} \times 2n} - S$ is rigid if and only if $(K'_{\frac{k+d-2}{2} \times 2n} - S) + F$ is rigid.

Now $|V_{i_1} - S| \leq d-1$ and $|V_{i_2} - S| \leq d-1$ imply that $|V_{i_1} \cap S| \geq \frac{k+d-2}{2} - (d-1) = \frac{k-d}{2}$ and $|V_{i_2} - S| \geq \frac{k-d}{2}$. Hence $|S \cap V_i| \leq d-1$ and hence $|V_i - S| \geq \frac{k-d}{2}$ holds for each index $i_1, i_2 \neq i \in \{1, \ldots, n\}$. Like in the previous case, for each $i_1 < i < i_2$, we denote the first $\frac{k-d}{2}$ elements of $V_i - S$ by $\{v_1^i, \ldots, v_{\frac{k-d}{2}}^i\}$ and delete the rest of its elements since we may add them by 0-extensions in the end of the construction. Next we contract the vertex sets $\{v_1^{i_1+1}, v_1^{i_1+3}, \ldots, v_1^{i_1+2\lfloor \frac{i_2-i_1}{2} \rfloor - 1}\}, \ldots, \{v_{\frac{k-d}{2}}^{i_1+1}, v_{\frac{k-d}{2}}^{i_1+3}, \ldots, v_{\frac{k-d}{2}}^{i_1+2\lfloor \frac{i_2-i_1}{2} \rfloor - 1}\}$ and $\{v_1^{i_1+2}, v_1^{i_1+4}, \ldots, v_1^{i_1+2\lceil \frac{i_2-i_1}{2} \rceil - 2}\}, \ldots, \{v_{\frac{k-d}{2}}^{i_1+2}, v_{\frac{k-d}{2}}^{i_1+4}, \ldots, v_{\frac{k-d}{2}}^{i_1+2\lceil \frac{i_2-i_1}{2} \rceil - 2}\}$ in $(K'_{\frac{k+d-2}{2} \times 2n} - S) + F$ into single vertices $u_1^1, \ldots, u_{\frac{k-d}{2}}^1$ and $u_1^2, \ldots, u_{\frac{k-d}{2}}^2$. Recall that we are assuming now that $i_2 - i_1 \geq 3$ and hence we indeed get at least $2\frac{k-d}{2}$ contracted vertices.) Let H be the resulting graph.

We have now two cases. Assume first that $|V_{i_1} - S| + |V_{i_2} - S| \geq d$. Let now $H' = H - (V_1 \cup \cdots \cup V_d \cup V_{d+1} \cup \cdots \cup V_{i_1-1} \cup V_{i_2+1} \cup \cdots \cup V_{2n} - S)$. Observe that H' is the union of the complete graph $K_{(V_{i_1} \cup V_{i_2}) - S}$ and either the complete bipartite graph $K_{\frac{k-d}{2}+|V_{i_1}-S|, \frac{k-d}{2}+|V_{i_2}-S|}$ or $K_{\frac{k-d}{2}+|V_{i_1}-S|+|V_{i_2}-S|, \frac{k-d}{2}}$. Since we assumed that $|V_{i_1} - S| + |V_{i_2} - S| \geq d$, H' is rigid by Lemma 10 and hence we can conclude that H is rigid by using 0-extensions. Next assume that $|V_{i_1} - S| + |V_{i_2} - S| = d-1$ (which is the single case left). Let now $H'' = H/(V_1 \cup \cdots \cup V_d \cup V_{d+1} \cup \cdots \cup V_{i_1-1} \cup V_{i_2+1} \cup \cdots \cup V_{2n} - S)$, that is, the graph that we get from H after contracting the vertex set $V_1 \cup \cdots \cup V_d \cup V_{d+1} \cup \cdots \cup V_{i_1-1} \cup V_{i_2+1} \cup \cdots \cup V_{2n} - S$ into a single vertex v_0. Observe that H'' is the union of the complete graph $K_{((V_{i_1} \cup V_{i_2}) - S) \cup \{v_0\}}$ and either the complete bipartite graph $K_{\frac{k-d}{2}+|V_{i_1}-S|+1, \frac{k-d}{2}+|V_{i_2}-S|}$ or $K_{\frac{k-d}{2}+|V_{i_1}-S|+|V_{i_2}-S|+1, \frac{k-d}{2}}$. Since $|V_{i_1} - S| + |V_{i_2} - S| = d-1$, H'' is rigid by Lemma 10. Now, we can conclude that H is rigid by using Theorem 6 for splitting v_0 multiple times, (and using the edges from v_0 to $(V_{i_1} \cup V_{i_2}) - S$ as the designated edges).

As H is rigid, $K'_{\frac{k+d-2}{2} \times 2n} - S + F$ is also rigid by Theorem 7 (and 0-extensions). This, as we have seen earlier, implies the rigidity of $K'_{\frac{k+d-2}{2} \times 2n} - S$, finishing our proof. □

Other variants. Jordán [10] observed that a $[k,d]$-rigid graph is also $[k,d]$-edge rigid and $[k-1,d]$-vertex globally rigid. Similarly, a $[k-1,d]$-vertex globally rigid graph is also $[k-1,d]$-edge globally rigid. These facts along with Theorems 11 and 13 provide the following corollaries which imply that $K_{\frac{k+d-1}{2} \times n}$ and $K'_{\frac{k+d-2}{2} \times 2n}$ are extremal for the bounds of Theorem 2.

Corollary 14 *Assume that $d \geq 2$, $n \geq 5d+1$, $k \geq \frac{d^2}{4} + 2d - 1$, and $k+d$ is odd. Then $K_{\frac{k+d-1}{2} \times n}$ is $[k,d]$-edge rigid, $[k-1,d]$-vertex globally rigid, and $[k-1,d]$-edge globally rigid.* □

Corollary 15 *Assume that $d \geq 2$, $2n \geq 5d+1$, $k \geq \frac{d^2}{4} + 2d + 2\}$, and $k+d$ is even. Then $K'_{\frac{k+d-2}{2} \times 2n}$ is $[k,d]$-edge rigid, $[k-1,d]$-vertex globally rigid, and $[k-1,d]$-edge globally rigid.* □

4 Upper bounds

We show here how the techniques of [15] can be used to prove an almost tight upper bound for the edge-redundant version of Theorems 3 and 5, verifying (an extension of) the conjecture of [4]. The key of our proofs is the following observation.

Lemma 16 *Let $G=(V,E)$ be a simple graph and $F \subset E$ be an edge set with $|F| = k-1$. Suppose that $e \in E-F$ is an $\mathcal{R}_d$-bridge of $G-F_e$. Then e is an $\mathcal{R}_{d+k-1}$-bridge of G.*

Proof: Let $F = \{u_1v_1, ..., u_{k-1}v_{k-1}\}$. By possibly changing the role of u_i and v_i, we may assume that no v_i is an end vertex of e (since G is simple). As e is an $\mathcal{R}_d$-bridge of $G-F$, it is also an $\mathcal{R}_d$-bridge of $G-\{v_1,\dots,v_{k-1}\} \subset G-F_e$. By [15, Lemma 3], e is an $\mathcal{R}_{d+k-1}$-bridge of $(\dots(G-\{v_1,\dots,v_{k-1}\})*v_1)*\dots)*v_{k-1}$, where $G*v$ is the cone of G that we get from G by adding a new vertex v and connecting it to each vertex of G; and here we add the copies of the same vertex (if $v_i = v_j$) multiple times. Hence e is an $\mathcal{R}_{d+k-1}$-bridge of G as it is a subgraph of $(\dots(G-\{v_1,\dots,v_{k_1}\})*v_1)*\dots)*v_{k-1}$. □

Theorem 17 *Let $G=(V,E)$ be a minimally $[k,d]$-edge rigid simple graph on at least $d+k$ vertices. Then*

$$|E| \leq (d+k-1)|V| - \binom{d+k}{2}.$$

Proof: As $G-e$ is not $[k,d]$-edge rigid for each edge e, there is a set $F_e \subseteq E$ such that $|F_e| = k-1$, $G-F_e-e$ is not rigid. On the other hand, $G-F_e$ is rigid by the $[k,d]$-edge rigidity of G and hence e is an $\mathcal{R}_d$-bridge of $G-F_e$. Thus Lemma 16 implies that e is is an $\mathcal{R}_{d+k-1}$-bridge of G for each edge e. Now, the upper bound on $|E|$ follows by the fact that the maximum rank of the d-dimensional rigidity matrix is $d|V| - \binom{d+1}{2}$ if $|V| \geq d+1$. □

Theorem 18 *Let $G=(V,E)$ be a minimally $[k,d]$-edge globally rigid simple graph on at least $d+k+1$ vertices where $d=1,2$. Then*

$$|E| \leq (d+k)|V| - \binom{d+k+1}{2}.$$ □

Proof: As $G-e$ is not $[k,d]$-edge globally rigid for each edge e, there is a set $F_e \subseteq E$ such that $|F_e| = k-1$, $G-F_e-e$ is not globally rigid. On the other hand, $G-F_e$ is globally rigid by the $[k,d]$-edge global rigidity of G and hence e is an $\mathcal{R}_{d+1}$-bridge of $G-F_e$ by Theorem 4. Thus Lemma 16 implies that e is is an $\mathcal{R}_{d+k}$-bridge of G for each edge e. Now, the upper bound on $|E|$ follows by the fact that the maximum rank of the d-dimensional rigidity matrix is $d|V| - \binom{d+1}{2}$ if $|V| \geq d+1$. □

The above bounds are tight for $|V| = d+k$ ($|V| = d+k+1$, respectively) by the minimal $[k,d]$-edge rigidity ($[k,d]$-edge global rigidity, respectively) of the complete graph K_{d+k} (K_{d+k+1}, respectively). For higher values of $|V|$, the bounds are almost tight as $K_{d+k-1,n-d-k+1}$ ($K_{d+k,n-d-k}$, respectively) is minimally $[k,d]$-edge rigid (globally rigid,respectively) for $n \geq \binom{d+2}{2}$. (The proofs of these facts are easy, so we leave them to the reader.)

5 Concluding remarks and open problems

We have shown that the lower bound of the number of edges in $[k,d]$-rigid, $[k,d]$-edge rigid, $[k,d]$-vertex globally rigid, and $[k,d]$-edge globally rigid graphs provided by Theorems 1 and 2 are tight for all $d \geq 2$ when k is sufficiently large. The conjecture of [15] that these lower bounds are also tight for all k with $k \geq d+2$ remains open for $k \leq \frac{d^2}{4}$. We note that, when $k \leq d+1$, the problem of giving tight lower bounds to the above types of graphs is more challenging since in these cases (usually) no regular graphs will be sufficient for the proof. Table 1 summarizes the current knowledge on these types of problems.

$k=$	1	2	3	4	$> d+1$
$[k,2]$-r	$2n-3$	$2n-1$ [20]	$2n+2$ [13, 19]	$\lceil \frac{5n}{2} \rceil$ [10]	$\lceil \frac{(k+1)n}{2} \rceil$ [10]
$[k,2]$-e-r	$2n-3$	$2n-2$	$2n$ [10]	$\lceil \frac{5n}{2} \rceil$ [10]	$\lceil \frac{(k+1)n}{2} \rceil$ [10]
$[k,2]$-g-r	$2n-2$	$2n$ [19, 24]	$\lceil \frac{5n}{2} \rceil$ [24]	$3n$ [10]	$\lceil \frac{(k+2)n}{2} \rceil$ [10]
$[k,2]$-e-g-r	$2n-2$	$2n$ [10]	$\lceil \frac{5n}{2} \rceil$ [10]	$3n$ [10]	$\lceil \frac{(k+2)n}{2} \rceil$ [10]
$[k,3]$-r	$3n-6$	$3n-3$ [15]	$3n$ [15]	$3n+5+\varepsilon$ [14]	$\lceil \frac{(k+2)n}{2} \rceil$ [10]
$[k,3]$-e-r	$3n-6$	$3n-5$	$3n-4$ [14]	$3n$ [14]	$\lceil \frac{(k+2)n}{2} \rceil$ [14]
$[k,3]$-g-r	$3n-5$ [10]	$3n-2$[17]	$3n+2+\delta$ [14]	$\lceil \frac{7n}{2} \rceil$ [14]	$\lceil \frac{(k+3)n}{2} \rceil$ [14]
$[k,3]$-e-g-r	$3n-5$ [10]	$3n-4$ [2]	$3n$ [14]	$\lceil \frac{7n}{2} \rceil$ [14]	$\lceil \frac{(k+3)n}{2} \rceil$ [14]
$[k,d]$-r	$dn-\binom{d+1}{2}$	$dn-\binom{d}{2}$ [15]	OPEN for $d \geq 4$	OPEN for $d \geq 4$	$\boldsymbol{\lceil \frac{(k+d-1)n}{2} \rceil^*}$
$[k,d]$-e-r	$dn-\binom{d+1}{2}$	$dn-\binom{d+1}{2}+1$	OPEN for $d \geq 4$	OPEN for $d \geq 4$	$\boldsymbol{\lceil \frac{(k+d-1)n}{2} \rceil^*}$
$[k,d]$-g-r	$dn-\binom{d+1}{2}+1$ [10]	OPEN for $d \geq 4$	OPEN for $d \geq 4$	OPEN for $d \geq 4$	$\boldsymbol{\lceil \frac{(k+d)n}{2} \rceil^*}$
$[k,d]$-e-g-r	$dn-\binom{d+1}{2}+1$ [10]	OPEN for $d \geq 4$	OPEN for $d \geq 4$	OPEN for $d \geq 4$	$\boldsymbol{\lceil \frac{(k+d)n}{2} \rceil^*}$

Table 1: The known tight lower bounds on the edge number, where $0 \leq \varepsilon \leq 15$ and $0 \leq \delta \leq 18$. The bold bounds are from this paper and are only known to be tight for sufficiently large k.

We also provided upper bounds for the size of minimally $[k,d]$-edge rigid ($[k,d]$-edge globally, respectively) simple graphs for all d (for $d=1,2$, respectively). We have seen that the tightness of these bound was only provided for $|V| = k+d$ ($|V| = k+d+1$, respectively), however, for higher values of $|V|$, we only gave almost tight examples. Based on the results of Jordán [11] on the $[2,2]$-edge rigidity case, we conjecture that the tight upper bounds (for sufficiently large n) coincide with the edge number of $K_{k+d-1,n-k-d+1}$ ($K_{k+d-1,n-k-d+1}$, respectively). A rather important problem is the extension of Theorem 4 for higher values of d, that is, the verification of the conjecture of [9, 16]. Such result will imply that Theorems 5 and 18 are also true for higher values of d.

Acknowledgments. Projects nos. NKFI-128673 and PD-138102 have been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the FK_18 and PD_21 funding schemes. This paper was supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences and by the ÚNKP-20-5 and ÚNKP-21-5 New National Excellence Program of the Ministry for Innovation and Technology.

References

[1] L. Berenchtein, L. Chavez, and W. Whiteley. Inductive constructions for 2-rigidity: bases and circuits via tree partitions. Manuscript, York University, Toronto, 2002.

[2] Q. Chen, S. Jajodia, T. Jordán, and K. Perkins. Redundantly globally rigid braced triangulations. Technical Report TR-2021-12, Egerváry Research Group, Budapest, 2021. `egres.elte.hu`.

[3] R. Connelly. Generic global rigidity. *Discrete & Computational Geometry*, 33(4):549–563, 2005.

[4] D. Garamvölgyi and T. Jordán. Minimally globally rigid graphs. *Eur. J. Comb.*, 108:103626, 2023.

[5] S.J. Gortler, A.D. Healy, and D.P. Thurston. Characterizing generic global rigidity. *American Journal of Mathematics*, 132(4):897–939, 2010.

[6] J.E. Graver, B. Servatius, and H. Servatius. *Combinatorial Rigidity*. AMS Graduate studies in mathematics Vol. 2. American Mathematical Soc., 1993.

[7] B. Hendrickson. Conditions for unique graph realizations. *SIAM J. Comput.*, 21(1):65–84, 1992.

[8] T. Jordán. Combinatorial rigidity: Graphs and matroids in the theory of rigid frameworks. In *Discrete Geometric Analysis*, volume 34 of *MSJ Memoirs*, pages 33–112. Mathematical Society of Japan, Japan, 2016.

[9] T. Jordán. Extremal problems and results in combinatorial rigidity. In A. Frank, A. Recski, and G. Wiener, editors, *Proc. of the 10th Japanese-Hungarian Symposium on Discrete Mathematics and Its Applications May 22-25, 2017, Budapest, Hungary*, pages 297–303. Department of Computer Science and Information Theory, Budapest University of Technology and Economics, 2017.

[10] T. Jordán. Minimum size highly redundantly rigid graphs in the plane. *Graphs Comb.*, 37(4):1415–1431, 2021.

[11] T. Jordán. Ear-decompositions, minimally connected matroids, and rigid graphs. Technical Report TR-2022-11, Egerváry Research Group, Budapest, 2022. `egres.elte.hu`.

[12] T. Jordán. The globally rigid complete bipartite graphs. Technical Report (Quick Proof) QP-2022-02, Egerváry Research Group, Budapest, 2022. `egres.elte.hu`.

[13] T. Jordán, R. Huang, H. Simmons, K. Weatherspoon, and Z. Zheng. Four-regular graphs with extremal rigidity properties. Technical Report TR-2022-13, Egerváry Research Group, Budapest, 2022. `egres.elte.hu`.

[14] T. Jordán, C. Poston, and R. Roach. Extremal families of redundantly rigid graphs in three dimensions. *Discret. Appl. Math.*, 322:448–464, 2022.

[15] V.E. Kaszanitzky and Cs. Király. On minimally highly vertex-redundantly rigid graphs. *Graphs Comb.*, 32(1):225–240, 2016.

[16] Cs. Király. *Graph Structures from Combinatorial Optimization and Rigidity Theory*. PhD thesis, ELTE, Budapest, 2015.

[17] Cs. Király. Unpublished result, 2022.

[18] W. Mader. Über n-fach zusammenhängende, unendliche Graphen und ein extremal Problem. *Arch. Math.*, 23:553–60, 1972.

[19] S.A. Motevallian, C. Yu, and B.D.O. Anderson. On the robustness to multiple agent losses in 2d and 3d formations. *Int. J. of Robust and Nonlinear Control*, 2014.

[20] B. Servatius. Birigidity in the plane. *SIAM J. Discrete Math.*, 2(4):582–589, 1989.

[21] W. Whiteley. Infinitesimal motions of a bipartite framework. *Pacific J. Math.*, 110(1):233–255, 1984.

[22] W. Whiteley. Vertex splitting in isostatic frameworks. *Structutal Topology*, 16:22–30, 1990.

[23] W. Whiteley. Some matroids from discrete applied geometry. In J.E. Bonin, J.G. Oxley, and B. Servatius, editors, *Matroid Theory*, volume 197 of *Contemporary Mathematics*, pages 171–311. AMS, 1996.

[24] C. Yu and B.D.O. Anderson. Development of redundant rigidity theory for formation control. *International Journal of Robust and Nonlinear Control*, 19(13):1427–1446, 2009.

Scheduling under a resource constraint: the case of negligible processing times

Kristóf Bérczi[1]

Department of Operations Research
Eötvös Loránd University
Budapest, Hungary
kristof.berczi@ttk.elte.hu

Tamás Király[1]

ELKH–ELTE Egerváry Research Group
Department of Operations Research
Eötvös Loránd University
Budapest, Hungary
tamas.kiraly@ttk.elte.hu

Simon Omlor

Faculty of Statistics
TU Dortmund
Dortmund, Germany
simon.omlor@tu-dortmund.de

Abstract: We consider single-machine scheduling with a non-renewable resource. In this setting, we are given a set jobs, each characterized by a processing time, a weight, and a resource requirement. At fixed points in time, certain amounts of the resource are made available to be consumed by the jobs. The goal is to assign the jobs non-preemptively to time slots on the machine, so that each job has enough resource available at the start of its processing. In this talk, we consider the case when processing times are negligible, so every job can be scheduled at some resource arrival time.

Our main contribution is a PTAS for minimizing the weighted sum of completion times. We also investigate a variant where the resource arrival times are unknown, and present a $(4+\epsilon)$-approximation algorithm, together with a $(4-\varepsilon)$-inapproximability result for any $\varepsilon > 0$.

Keywords: scheduling; approximation; non-renewable resource; PTAS

1 Introduction

Scheduling problems with non-renewable resource constraints arise naturally in various areas where resources like raw materials, energy, or financial funding arrive at predetermined dates. In the general setting, we are given a set of jobs and a set of machines. Each job is equipped with a requirement vector that encodes the needs of the given job for the different types of resources. There is an initial stock for each resource, and some additional resource arrival times in the future are known together with the arriving quantities. The aim is to find a schedule of the jobs on the machines such that the necessary resources are available for each job.

In the present talk, we concentrate on the problem with a single resource, where the objective is to minimize the weighted sum of completion times. Furthermore, we assume that every job has 0 processing time. This means that the number of machines is irrelevant, and each job can be scheduled at a resource arrival time. This case is relevant to situations where processing times are negligible compared to the gaps

[1]Research is supported by supported by the Lendület Programme of the Hungarian Academy of Sciences – grant number LP2021-1/2021, and by the Hungarian National Research, Development and Innovation Office – NKFIH, grant number FK128673

between resource arrival times, and the bottleneck is resource availability. Examples include financial scheduling problems where the jobs are not time-consuming but the availability of funding varies in time, or production problems where products are shipped at fixed time intervals and production time is negligible compared to these intervals. Note that the number of machines is irrelevant if processing times are 0.

Related work Scheduling problems with resource restrictions (also called financial constraints, or raw material restrictions) were introduced by Carlier and Rinnooy Kan [2] and by Slowinski [7]. Carlier [1] settled the computational complexity of several variants for the single machine case. In particular, he showed that the problem of minimizing the weighted sum of completion times is NP-hard.

Kis [6] showed that the problem remains weakly NP-hard even when the resource arrives at only 2 distinct dates. On the positive side, he gave an FPTAS for this case.

Györgyi and Kis [4] gave polynomial-time algorithms for several special cases, and also showed that the problem remains weakly NP-hard under the very strong assumption that for each job, the processing time, resource requirement and weight are the same. They also provided a 2-approximation algorithm for this variant, and a polynomial-time approximation scheme (PTAS) when the number of resource arrival times is a constant and the processing time equals the weight for each job, while the resource requirements are arbitrary. The same authors showed in [5] that minimizing the sum of completion times is NP-hard even for two resource arrival times and all jobs having unit resource requirement, and provided a FPTAS for a variant in which the jobs have arbitrary weights, but the resource requirements are identical and the number of resource arrival times is bounded by a constant.

Notation Throughout the paper, we will use the following notation. We are given a set J of n jobs. Each job $j \in J$ has a non-negative weight w_j and a resource requirement a_j. The resources arrive at time points $t_1, \ldots, t_q$, and the amount of resource that arrives at t_i is denoted by b_i. We assume that $\sum_{i=1}^{q} b_i = \sum_{j=1}^{n} a_j$ holds. We also assume that $t_1 = 0$ – this is a valid assumption because it is the worst case for the approximation ratio.

Since the processing times are 0, every job is processed at one of the arrival times in any optimal schedule. Thus, a *schedule* can be represented by a mapping $\pi : J \to [q]$, where $\pi(j)$ denotes the index of the resource arrival time when job j is processed. The *completion time* C_j of job j equals $t_{\pi(j)}$.

A schedule is feasible if the resource requirements are met, that is, if

$$\sum_{j:\pi(j)\le k} a_j \le \sum_{i\le k} b_i \tag{1}$$

for all $1 \le k \le q$. Since we assume that $\sum_i b_i = \sum_j a_j$ holds, this is equivalent to

$$\sum_{j:\pi(j)\ge k} a_j \ge \sum_{i\ge k} b_i \tag{2}$$

for all $1 \le k \le q$. Define $B_k = \sum_{i\ge k} b_i$, and consider the set of jobs that are not processed before a given time point t_i. Inequality (2) means that if the resource requirements of these jobs add up to at least B_i, then our schedule is feasible.

Using the standard $\alpha|\beta|\gamma$ notation of Graham, Lawler, Lenstra and Kan [3], our problem is denoted by $1|rm = 1, p_j = 0|\sum C_j w_j$, where rm stands for *raw materials*, and $p_j = 0$ means that the processing time is 0 for every job.

2 Our results

We consider problem $1|rm = 1, p_j = 0|\sum C_j w_j$. The problem clearly is NP-hard even for $q = 2$, as the knapsack problem can be reduced to it. Indeed, maximizing the weight of the items in the knapsack is

equivalent to the task of maximizing the weight of jobs that are scheduled at the first resource arrival time.

First, as an introductory result, we show a $(4+\varepsilon)$-approximation algorithm that serves as a simplified illustration of the technique used in the general PTAS.

As a next step towards a general PTAS, we present a PTAS for constant number of resource arrival times. This uses a fairly standard method of guessing the k heaviest jobs at each arrival time, and it will be used as a subroutine in our algorithm for the general case. We give a $(1+\frac{q}{k})$-approximation algorithm with running time $\mathcal{O}(n^{qk+1})$, where q is the number of arrival times. Then we prove the main result, which is a PTAS for an arbitrary number of resource arrival times.

Finally, we consider a variant of the problem where the number of arrival times and the arriving quantities are known, but the arrival times themselves are unknown. On one hand, we show that no $(4-\varepsilon)$-approximation algorithm is possible for any $\varepsilon > 0$. On the other hand, we give a $(4+\varepsilon)$-approximation algorithm with running time polynomial in $1/\varepsilon$ and the input length.

2.1 A $(4+\varepsilon)$-approximation for arbitrary q

We may assume without loss of generality that resource arrival times are integer. The idea of the algorithm is as follows. First, we shift all resource arrival times to powers of 2. For each time point t_i $(i > 1)$ in the shifted instance, we apply the FPTAS by Kis [6] to the instance which has only two resource arrival times t_1 and t_i, and the resource quantity for t_i is B_i.

Denote the set of jobs assigned to t_i this way by S_i. Then, going backwards from the last time point t_q to the first one t_1, we assign all previously unassigned jobs from S_i to t_i, i.e. $\pi(j) = \max\{i : \ j \in S_i\}$.

More formally, let $\mathcal{I}$ be an instance of $1|rm = 1, p_j = 0|\sum_j C_j w_j$. We assume $t_1 = 0$ and $t_2 = 1$ (the latter can be assumed because we can add a dummy arrival time with $b_2 = 0$). We define a new instance $\mathcal{I}'$ of $1|rm = 1, p_j = 0|\sum_j C_j w_j$ with shifted resource arrival times as follows. Let $a'_j = a_j$ and $w'_j = w_j$ for every $j \in J$, and let

$$t'_i = \begin{cases} 0 & \text{if } i = 1, \\ 2^{i-2} & \text{for } i = 2, \ldots, \lceil \log_2(t_q) \rceil + 2, \end{cases}$$

$$b'_i = \begin{cases} b_i & \text{if } i = 1, 2 \\ \sum [b_\ell : \ t_\ell \in (2^{i-3}, 2^{i-2}] \ ] & \text{for } i = 3, \ldots, \lceil \log_2(t_q) \rceil + 2. \end{cases}$$

Claim 1 *A solution to $\mathcal{I}$ with weighted sum of completion times W can be transformed into a solution of $\mathcal{I}'$ with weighted sum of completion times at most $2W$. Furthermore, any feasible schedule for $\mathcal{I}'$ is also feasible for $\mathcal{I}$.*

Proof: Let us define $t^*_i = \min\{t'_\ell : \ t_i \le t'_\ell\}$ for $i = 1, \ldots, q$. Let π be the solution for $\mathcal{I}$. Then assigning all jobs that are assigned to time point t_i to t^*_i gives us a feasible solution to $\mathcal{I}'$. By this change, the completion time of any job is at most doubled (recall that each t_i is assumed to be integer).

Since the available amount of resources at each time in $\mathcal{I}'$ is at most as much as in $\mathcal{I}$, a feasible schedule for $\mathcal{I}'$ is also a feasible schedule for $\mathcal{I}$. □

Claim 2 *For instances $\mathcal{I}$ where the resource arrival times are integer powers of 2, there exists a $(2+\varepsilon)$-approximation algorithm with running time polynomial in the input size and $1/\varepsilon$.*

Proof: We use the procedure that we described above, i.e., for each $i > 1$ we solve the instance with B_i resource arriving at t_i and the rest at t_1, using the FPTAS provided by [6]. As defined above, S_i is the set of jobs assigned to t_i by the FPTAS, and $\pi(j) = \max\{i : \ j \in S_i\}$.

Let π^{opt} be an optimum solution and let J_k^{opt} be the set of jobs j with $\pi^{\mathsf{opt}}(j) = k$. We have

$$w(S_i) \le (1+\varepsilon) \sum_{k=i}^{q} w(J_k^{\mathsf{opt}})$$

for $i = 1, \ldots, q$. Then we get

$$2(1+\varepsilon)\sum_{j\in J} w_j C_j^{\pi^{\mathsf{opt}}} = \sum_{i=2}^{q} 2(1+\varepsilon)\cdot 2^{i-2} w(J_i^{\mathsf{opt}}) = \sum_{i=2}^{q} (1+\varepsilon)\cdot 2^{i-2} w(J_i^{\mathsf{opt}}) \left(1 + \sum_{j=1}^{\infty} 2^{-j}\right)$$
$$\geq \sum_{i=2}^{q} (1+\varepsilon) 2^{i-2} \sum_{k=i}^{q} w(J_k^{\mathsf{opt}}) \geq \sum_{i=2}^{q} 2^{i-2} w(S_i),$$

thus the approximation ratio follows. □

The two claims show that this approach leads to a $(4+\varepsilon)$-approximation with running time polynomial in $1/\varepsilon$ and the input size.

2.2 PTAS for constant q

In this section we give a PTAS for the case when the number of resource arrival times is constant. Recall that Kis [6] gave a FPTAS for $1|rm = 1|\sum C_j w_j$ when there are two resource arrival times.

Our algorithm is a generalization of a well-known PTAS for the knapsack problem, and will be used later as a subroutine in the PTAS for an arbitrary number of resource arrival times. The idea is to choose a number $k \in \mathbb{Z}_+$, guess the k heaviest jobs that are processed at each resource arrival time t_i, and then determine the remaining jobs that are scheduled at t_i in a greedy manner. Since we go over all possible sets containing at most k jobs for each resource arrival time, there is an exponential dependence on the number q of resource arrival times in the running time.

Theorem 3 *There is a $(1 + \frac{q}{k})$-approximation algorithm with running time $\mathcal{O}(n^{qk+1})$, where q is the number of arrival times.*

Algorithm 1 PTAS for $1|rm = 1, p_j = 0|\sum C_j w_j$ when q is a constant.

Input: Jobs J with $|J| = n$, resource requirements a_j, weights w_j, resource arrival times $t_1 \leq \ldots \leq t_q$ and resource quantities $b_1, \ldots b_q$.
Output: A feasible schedule π.

for all subpartitions $S_1 \cup \cdots \cup S_q \subseteq J$ with $|S_i| \leq k$ for $i > 1$ **do**
  Set $A = 0$.
  Set $W = 0$.
  for i from 0 to $q - 2$ **do**
    for $j \in S_{q-i}$ **do**
      $\pi(j) = q - i$
      $A \leftarrow A + a_j$
    if $|S_{q-i}| = k$ **then**
      $W \leftarrow \max\{W, \min\{w_j : \ j \in S_{q-i}\}\}$
      while $A < B_{q-i}$ **do**
        if there exists an unassigned job j with $w_j \leq W$ **then**
          Let j be an unassigned job with $w_j \leq W$ minimizing w_j/a_j.
          $\pi(j) = q - i$
          $A \leftarrow A + a_j$
        else
          break
  For all remaining jobs set $\pi(j) = 1$.
Let π be the best schedule found.
return π

PROOF: We claim that Algorithm 1 satisfies the requirements. Let π^{opt} be an optimal schedule and define $J_i^{\mathsf{opt}} = \{j \in J : \pi^{\mathsf{opt}}(j) = i\}$. Let S_i^{opt} be the set of the k heaviest jobs in J_i^{opt} if $|J_i^{\mathsf{opt}}| \geq k$, otherwise let $S_i^{\mathsf{opt}} = J_i^{\mathsf{opt}}$. Let $J_i = \{j \in J : \pi(j) = i\}$ denote the set of jobs assigned to time t_i in our solution. In each iteration of the *for* loop of Step 4, let j_i be the last job added to J_i if such a job exists.

Assume that we are at the iteration of the algorithm when the subpartition $S_1^{\mathsf{opt}} \cup \cdots \cup S_q^{\mathsf{opt}}$ is considered in Step 1. Let $W_{q-\ell}$ denote the value of W at the end of the iteration of the for loop corresponding to $i = \ell$ in Step 4. To show feasibility of π, observe that any job $j \notin S_1^{\mathsf{opt}} \cup \cdots \cup S_q^{\mathsf{opt}}$ for which $\pi^{\mathsf{opt}}(j) \geq q - \ell$ always satisfies $w_j \leq W_{q-\ell}$, so we can pick jobs in line 11 until $A \geq B_{q-\ell}$.

Now we prove the approximation factor. By Steps 3 and 9, we have

$$W_{q-\ell} \leq \frac{1}{k} \sum_{i=\ell}^{q} \sum_{j \in J_i^{\mathsf{opt}}} w_j.$$

As our algorithm always picks the most inefficient job, we also have

$$\sum_{i=\ell}^{q} \sum_{j \in J_i \setminus \{j_i\}} w_j \leq \sum_{i=\ell}^{q} \sum_{j \in J_i^{\mathsf{opt}}} w_j,$$

where $J_i \setminus \{j_i\} = J_i$ if j_i is not defined for i. Combining these two observations, for $\ell = 1, \ldots, q$ we get

$$\sum_{i=\ell}^{q} \sum_{j \in J_i} w_j = \sum_{i=\ell}^{q} \sum_{j \in J_i \setminus \{j_i\}} w_j + \sum_{i=\ell}^{q} w_{j_i} \leq \sum_{i=\ell}^{q} \sum_{j \in J_i^{\mathsf{opt}}} w_j + (q - \ell + 1) \cdot W_\ell \leq (1 + \frac{q}{k}) \sum_{i=\ell}^{q} \sum_{j \in J_i^{\mathsf{opt}}} w_j,$$

where the first inequality follows from the fact that $w_{j_i} \leq W_i \leq W_\ell$ whenever $i \geq \ell$. This proves that the schedule that we get is a $(1 + \frac{q}{k})$-approximation.

We get a factor of n^{qk} in the running time for guessing the sets S_k. Assigning the remaining jobs can be done in linear time by ordering the jobs and using AVL-trees, thus we get an additional factor of n. In order to get a PTAS, we set $k = \frac{\varepsilon}{q}$. $\square$

2.3 PTAS for arbitrary q

We turn to the proof of the main result of the paper. As in Section 2.1, we shift resource arrival times; here we use powers of $1 + \varepsilon$, for a suitably small ε.

Let $\mathcal{I}$ be an instance of $1|rm = 1, p_j = 0| \sum_j C_j w_j$. We assume that resource arrival times are integer, and that $t_1 = 0$, $t_2 = 1$. We define a new instance $\mathcal{I}'$ of $1|rm = 1, p_j = 0| \sum_j C_j w_j$ with shifted resource arrival times as follows. Let $a'_j = a_j$ and $w'_j = w_j$ for every $j \in J$, and set

$$t'_i = \begin{cases} 0 & \text{if } i = 1, \\ (1+\varepsilon)^{i-2} & \text{for } i = 2, \ldots, \lceil \log_{1+\varepsilon}(t_q) \rceil + 2, \end{cases}$$

$$b'_i = \begin{cases} b_i & \text{if } i = 1, 2, \\ \sum [b_\ell : \ t_\ell \in ((1+\varepsilon)^{i-3}, (1+\varepsilon)^{i-2}] \ ] & \text{for } i = 3, \ldots, \lceil \log_{1+\varepsilon}(t_q) \rceil + 2. \end{cases}$$

The proof of the following claim is the same as that of Claim 1.

Claim 4 *A solution to $\mathcal{I}$ with weighted sum of completion times W can be transformed into a solution of $\mathcal{I}'$ with weighted sum of completion times at most $(1+\varepsilon)W$. Furthermore, any feasible schedule for $\mathcal{I}'$ is also a feasible schedule for $\mathcal{I}$.*

Due to the claim, we may assume that the positive arrival times are powers of $1 + \varepsilon$. For convenience of notation, in this section we will assume that the largest arrival time is 1, and arrival times are indexed in decreasing order, starting with $t_0 = 1$. That is, $t_i = (1 + \varepsilon)^{-i}$ $(i = 0, \ldots, q - 2)$, and $t_{q-1} = 0$. We will also assume that for a given constant r, $b_{q-r-1} = \cdots = b_{q-2} = 0$. This can be achieved by adding r dummy arrival times.

Theorem 5 *There exists a PTAS for* $1|rm = 1, p_j = 0|\sum C_j w_j$.

PROOF: Let us fix an even integer r and $\varepsilon > 0$; we will later assume that r is very large compared to ε^{-1}. We assume that resource arrival times are as described above, and are indexed in decreasing order.

In the algorithm, we fix jobs at progressively decreasing arrival times, by using the PTAS of the previous section for $r+1$ arrival times (except for the first step, when we may use the PTAS for less than $r+1$ arrival times). We will run our algorithm $r/2$ times with slight modifications, and pick the best result. Each run is characterized by a parameter $\ell \in \{1, \ldots, r/2\}$. See Algorithm 2.

Algorithm 2 PTAS for $1|rm = 1, p_j = 0|\sum C_j w_j$

Input: Jobs J with $|J| = n$, resource requirements a_j, weights w_j; an even integer r; resource quantities $b_0, \ldots b_{q-1}$ such that $b_{q-r-1} = \cdots = b_{q-2} = 0$ and $\sum a_j = \sum b_i$. We assume resource arrival times are $t_i = (1+\varepsilon)^{-i}$ $(i = 0, \ldots, q-2)$, $t_{q-1} = 0$.

Output: A feasible schedule π.

for ℓ from 1 to $r/2$ **do**
  Obtain instance $\mathcal{I}'$ with $r/2 + \ell + 1$ arrival times by moving arrivals before $t_{r/2+\ell-1}$ to 0
  Run Algorithm 1 on $\mathcal{I}'$ to get schedule σ.
  Let $A = B = 0$
  for i from 0 to $\ell - 1$ **do**
    For every $j \in \sigma^{-1}(i)$, fix $\pi_\ell(j) = i$
    $A \leftarrow A + \sum_{j \in \sigma^{-1}(i)} a_j$
    $B \leftarrow B + b_i$
  for j from 2 to $\lfloor 2(q-1-\ell)/r \rfloor$ **do**
    Let $s = (j-2)r/2 + \ell$
    Obtain instance $\mathcal{I}'$ with arrival times $t_s, t_{s+1}, \ldots, t_{s+r-1}, 0$: remove arrivals after t_s, remove $\max\{A - B, 0\}$ latest remaining resources, and move all arrivals before t_{s+r-1} to 0
    Let $A = B = 0$
    Run Algorithm 1 on $\mathcal{I}'$ to get schedule σ.
    for i from s to $s + r/2 - 1$ **do**
      For every $j \in \sigma^{-1}(i)$, fix $\pi_\ell(j) = i$
      $A \leftarrow A + \sum_{j \in \sigma^{-1}(i)} a_j$
      $B \leftarrow B + b_i$
  For all unscheduled jobs j, set $\pi_\ell(j) = q - 1$.
Let π be the best schedule among $\pi_1, \ldots, \pi_{r/2}$
return π

In the first step, we consider arrival times $t_0, t_1, \ldots, t_{r/2+\ell-1}, 0$. We move the resources arriving before $t_{r/2+\ell-1}$ to 0, and use the PTAS for $r/2 + \ell + 1$ arrival times on this instance. We fix the jobs that are scheduled at arrival times $t_0, t_1, \ldots, t_{\ell-1}$.

Consider now the jth step for some $j \geq 2$. Define $s = (j-2)r/2 + \ell$ and consider arrival times $t_s, t_{s+1}, \ldots, t_{s+r-1}, 0$. Move the resources arriving before t_{s+r-1} to 0, and decrease $b_s, b_{s+1}, \ldots$ in this order as needed, so that the total requirement of unfixed jobs equals the total resource. Use the PTAS for $r+1$ arrival times on this instance. Fix the jobs that are scheduled at arrival times $t_s, t_{s+1}, \ldots, t_{s+r/2-1}$.

The algorithm runs while $s + r - 1 \leq q - 2$, i.e., $jr/2 + \ell \leq q - 1$. Since the smallest r arrival times (except for 0) are dummy arrival times, the algorithm considers all resource arrivals.

The schedule given by the algorithm is clearly feasible, because when jobs at t_i are fixed, the total resource requirement of jobs starting no earlier than t_i is at least the total amount of resource arriving no earlier than t_i. To analyze the approximation ratio, we introduce the following notation: W_i is the total weight that the algorithm schedules at t_i; W'_i is the weight that the algorithm temporarily schedules at t_i when i is in the interval $[t_{s+r/2}, t_{s+r-1}]$ (or, in the first step, in the interval $[t_\ell, t_{\ell+r/2-1}]$); W^*_i is the total weight scheduled at t_i in the optimal solution.

Since we use the PTAS for $r/2+\ell+1$ arrival times in the first step, we have

$$\sum_{i=0}^{\ell-1}(1+\varepsilon)^{-i}W_i + \sum_{i=\ell}^{\ell+r/2-1}(1+\varepsilon)^{-i}W'_i \le (1+\varepsilon)\sum_{i=0}^{\ell+r/2-1}(1+\varepsilon)^{-i}W^*_i,$$

as the right-hand side is $(1+\varepsilon)$ times the objective value of the feasible solution obtained from the optimal solution by moving jobs arriving before $t_{\ell+r/2-1}$ to 0.

For $s = jr/2+\ell$, we compare the output of the PTAS with a different feasible solution: we schedule total weight W'_i at t_i for $i = s, s+1, \ldots, s+r/2-1$, total weight W^*_i at t_i for $i = s+r/2+1, \ldots, s+r-1$, and at $t_{s+r/2}$ we schedule all jobs that are no earlier than $t_{s+r/2}$ in the optimal schedule but are no later than $t_{s+r/2}$ in the PTAS schedule. We get the inequality

$$\begin{aligned}&\sum_{i=jr/2+\ell}^{(j+1)r/2+\ell-1}(1+\varepsilon)^{-i}W_i + \sum_{i=(j+1)r/2+\ell}^{(j+2)r/2+\ell-1}(1+\varepsilon)^{-i}W'_i \\ &\le (1+\varepsilon)\left(\sum_{i=jr/2+\ell}^{(j+1)r/2+\ell-1}(1+\varepsilon)^{-i}W'_i + \sum_{i=(j+1)r/2+\ell}^{(j+2)r/2+\ell-1}(1+\varepsilon)^{-i}W^*_i + (1+\varepsilon)^{-(j+1)r/2-\ell}\sum_{i=0}^{(j+1)r/2+\ell-1}W^*_i\right).\end{aligned}$$

The sum of these inequalities gives

$$\sum_{i=0}^{q-2}(1+\varepsilon)^{-i}W_i \le \varepsilon\sum_{i=\ell}^{q-2}(1+\varepsilon)^{-i}W'_i + (1+\varepsilon)\sum_{i=0}^{q-2}(1+\varepsilon)^{-i}W^*_i + (1+\varepsilon)\sum_{i=0}^{q-2}\left(\sum_{j:jr/2+\ell>i}(1-\varepsilon)^{-(jr/2+\ell)}\right)W^*_i. \tag{3}$$

To bound the first term on the right hand side of (3), first we observe that

$$\sum_{i=\ell}^{r/2+\ell-1}(1+\varepsilon)^{-i}W'_i \le (1+\varepsilon)\sum_{i=0}^{r/2+\ell-1}(1+\varepsilon)^{-i}W^*_i,$$

because the left side is at most the value of the PTAS in the first step, while the right side is $(1+\varepsilon)$ times the value of a feasible solution. Similarly,

$$\sum_{i=(j+1)r/2+\ell}^{(j+2)r/2+\ell-1}(1+\varepsilon)^{-i}W'_i \le (1+\varepsilon)\left(\sum_{i=jr/2+\ell}^{(j+2)r/2+\ell-1}(1+\varepsilon)^{-i}W^*_i + (1+\varepsilon)^{-jr/2-\ell}\sum_{i=0}^{jr/2+\ell-1}W^*_i\right),$$

because the left side is at most the value of the PTAS in the $(j+1)$-th step, and the right side is $(1+\varepsilon)$ times the value of the following feasible solution: take the optimal solution, move jobs scheduled before $t_{(j+2)r/2+\ell-1}$ to 0, and move jobs scheduled after $t_{jr/2+\ell}$ to $t_{jr/2+\ell}$. Adding these inequalities, we get

$$\begin{aligned}&\varepsilon\sum_{i=\ell}^{q-2}(1+\varepsilon)^{-i}W'_i \le \\ &\varepsilon(1+\varepsilon)\left(2\sum_{i=0}^{q-2}(1+\varepsilon)^{-i}W^*_i + \sum_{i=0}^{q-2}\left(\sum_{j:jr/2+\ell>i}(1+\varepsilon)^{-jr/2-\ell}\right)W^*_i\right) \le \\ &\varepsilon(1+\varepsilon)\left(2\sum_{i=0}^{q-2}(1+\varepsilon)^{-i}W^*_i + \sum_{i=0}^{q-2}\left(\sum_{j=0}^{\infty}(1+\varepsilon)^{-jr/2-1}\right)(1+\varepsilon)^{-i}W^*_i\right) = \\ &\varepsilon(1+\varepsilon)\left(2\sum_{i=0}^{q-2}(1+\varepsilon)^{-i}W^*_i + \frac{(1+\varepsilon)^{r/2-1}}{(1+\varepsilon)^{r/2}-1}\sum_{i=0}^{q-2}(1+\varepsilon)^{-i}W^*_i\right) = \\ &\varepsilon\left(2(1+\varepsilon) + \tfrac{(1+\varepsilon)^{r/2}}{(1+\varepsilon)^{r/2}-1}\right)\sum_{i=0}^{q-2}(1+\varepsilon)^{-i}W^*_i.\end{aligned}$$

The last expression is at most 4ε times the optimum value if r is large enough.

The last term of the right side of (3) is too large to get a bound that proves a PTAS. However, we can bound the *average* of these terms for different values of ℓ. The average is

$$(1+\varepsilon)\tfrac{2}{r}\sum_{\ell=1}^{r/2}\sum_{i=0}^{q-2}\left(\sum_{j:jr/2+\ell>i}(1-\varepsilon)^{-(jr/2+\ell)}\right)W_i^* \leq$$
$$(1+\varepsilon)\tfrac{2}{r}\sum_{i=0}^{q-2}\left(\sum_{j=1}^{\infty}(1+\varepsilon)^{-j}\right)(1-\varepsilon)^{-i}W_i^* = (1+\varepsilon)\frac{2}{r\varepsilon}\sum_{i=0}^{q-2}(1-\varepsilon)^{-i}W_i^*,$$

which is at most ε times the optimum if r is large enough. To summarize, we obtained that for large enough r, the average objective value of our algorithm for $\ell = 1, 2, \ldots, r/2$ is upper bounded by

$$4\varepsilon\sum_{i=0}^{q-2}(1+\varepsilon)^{-i}W_i^* + (1+\varepsilon)\sum_{i=0}^{q-2}(1+\varepsilon)^{-i}W_i^* + \varepsilon\sum_{i=0}^{q-2}(1+\varepsilon)^{-i}W_i^* = (1+6\varepsilon)\sum_{i=0}^{q-2}(1+\varepsilon)^{-i}W_i^*,$$

which is $(1+6\varepsilon)$ times the objective value of the optimal solution. This proves that the algorithm that chooses the best of the $r/2$ runs is a PTAS. □

2.4 Unknown resource arrival times

In this section we consider the variant of the problem where the arriving resource quantities b_j are known in advance, but the resource arrival times t_j are not. The problem is denoted by $1|rm = 1, p_j = 0, t_i \text{ unknown}|\sum C_j w_j$.

Recall that $B_k = \sum_{i \geq k} b_i$. Let J_i denote the minimum weight job set consuming at least B_i resources. It turns out that the question of approximability of the above problem can be reformulated as follows: Given $\alpha > 1$ and an instance of $1|rm = 1, p_j = 0|\sum_j C_j w_j$, is there a schedule π such that the set $S_i = \{j : \pi(j) \geq i\}$ has weight at most α times $w(J_i)$, for $i = 1, \ldots, q$? On one hand, such a solution would also give an α-approximation for the instance, since its objective value is at most α times the optimum for arbitrary resource arrival times. On the other hand, the smallest α for which the answer is affirmative in the above problem is the best approximation ratio we can achieve; this can be seen by choosing the i for which S_i is the worst approximation, and setting the arrival time t_j to 0 if $j < i$ and to 1 if $j \geq i$.

In the following, we show that for $\alpha > 4$ the answer to the above question is affirmative and a solution can be computed efficiently, while for $\alpha < 4$ there are instances where the answer is negative.

Theorem 6 *For $1|rm = 1, p_j = 0, t_i$ unknown$|\sum C_j w_j$, there exists a $(4+\varepsilon)$-approximation with running time polynomial in $1/\varepsilon$ and the input length. Moreover, there is no $(4-\varepsilon)$-approximation algorithm for the problem for any $\varepsilon > 0$.*

Proof: Our approximation algorithm is based on the following claim.

Claim 7 *There exists a schedule π such that for each i the set $S_i = \{j : \pi(j) \geq i\}$ is a 4-approximation for the problem of finding a minimum weight job set $S \subseteq J$ consuming at least B_i resources.*

Proof: Recall that J_i is a minimum weight job set consuming at least B_i resources. Define $f(i) = \min\{k : w(J_k) \leq 2w(J_i)\}$ for $i = 2, \ldots, q$ and let us consider the following procedure (Algorithm 3).

It is not difficult to see that π fulfills the resource requirements. We prove by induction that $w(S_i) \leq 4w(J_i)$ for $i = 1, \ldots, q$. As $S_q = J_{f(q)}$, the inequality $w(S_q) \leq 4w(J_q)$ clearly holds. Assume now that $i \leq q-1$. If no jobs are assigned to t_i, then $w(S_i) = w(S_{i+1}) \leq 4w(J_{i+1}) \leq 4w(J_i)$. Otherwise

Algorithm 3 Subroutine for $(4+\varepsilon)$-approximation to $1|rm=1, p_j=0, t_i \text{ unknown}|\sum C_j w_j$.

Input: Jobs J with $|J|=n$, resource requirements a_j, weights w_j, resource quantities $b_1, \dots b_q$.
Output: A feasible schedule π.

1: Set $i=q$.
2: **while** $i \geq 1$ **do**
3: Set $S_{i+1} = \{j : \pi(j) > i\}$.
4: Set $\pi(j) = i$ for $j \in J_{f(i)} \setminus S_{i+1}$.
5: $i \leftarrow f(i) - 1$
6: **return** π

$i = f(i')-1$, where i' is the index considered in the previous iteration of the while loop in Step 2. Observe that no jobs are assigned to time points between the ith and the i'th ones. By induction, we get

$$w(S_i) = w(J_{f(i)} \cup S_{i'}) \leq w(J_{f(i)}) + w(S_{i'}) \leq 2w(J_i) + 4w(J_{i'}) \leq 4w(J_i).$$

Here the second inequality holds by induction and by the definition of f, while the last inequality follows from the fact that $i < f(i')$ which implies $w(J_i) > 2w(J_{i'})$. □

Now we show how Algorithm 3 provides a $(4+\varepsilon)$-approximation for the problem $1|rm=1, p_j = 0, t_i \text{ unknown}|\sum C_j w_j$ which has running time polynomial in the input size and $\frac{1}{\varepsilon}$. Using either an FPTAS for the knapsack problem or the FPTAS of Kis [6], we determine an approximation of the sets J_i. Then we apply Algorithm 3 to schedule the jobs. This concludes the proof of the first part of the theorem.

The following set of instances shows that α is at least 4. We are given $(n-1)m$ jobs denoted by $1, 2, \dots, (n-1)m$ with weights $w_i = n - \frac{i}{m}$ and $a_i = 2^{-i}$. Furthermore, we have $(n-1)m$ resource arrival times that are unknown. The resource quantities are given by $b_q = 2^{-q}$ and $b_i = 2^{-i} - B_{i+1} = 2^{-i-1}$ for $1 < i < q$; b_1 equals the remaining resource requirement.

In order to fulfill the resource requirements, the set of jobs scheduled at or after time t_i has to contain at least one of the jobs $j \leq i$. Observe that the optimal solution of finding a minimum weight job set J_i consuming at least B_i resources consists of the single job i.

Let j_1 be a job processed at time t_q. We may assume that $w_{j_1} \leq 4w_q = 4$, because otherwise this schedule is not a 4-approximation if $t_q = 1$ and all other arrival times are set to 0.

We create a sequence starting with j_1. Since the jobs with indices greater or equal to j_1 have total resource requirement less than B_{j_1-1}, we have to schedule at least one job $j_2 < j_1$ at or after t_{j_1-1}. This argument can be iterated to find a job $j_3 < j_2$ which is scheduled at or after t_{j_2-1}, and so on, until $j_N = 1$ for some N.

The following claim shows that for n, m large enough, there must be an index i such that the jobs in this sequence that are scheduled at or after t_{j_i-1} have large total weight compared to $w_{j_i-1} = w(J_{j_i-1})$.

Claim 8 *For any $\beta < 4$, there exist n and m such that for any feasible schedule, there is some i with*

$$\beta w_{j_i-1} < \sum_{k=1}^{i+1} w_{j_k}$$

for the sequence $j_1, j_2, \dots$ constructed as above.

Proof: Suppose to the contrary that there is no triple n, m, i satisfying the requirements of the claim. Then for all n, m, there is a feasible schedule such that $\beta w_{j_i-1} \geq \sum_{k=1}^{i+1} w_{j_k}$ for all i.

By setting m large enough, $w_{j_i-1} - w_{j_i} = 1/m$ is very small; we will see that $m > 12/(4-\beta)$ is enough. By increasing n, the length N of our sequence increases as well. Indeed, by the indirect assumption, we have

$$w_{j_{i+1}} \leq \sum_{k=1}^{i+1} w_{j_k} \leq \beta w_{j_i-1} < 4w_{j_i-1} = 4w_{j_i} + 4/m.$$

Since we also have $w_{j_1} \leq 4$, $w_j \geq 1$ for all jobs j, and $w_1 = n$, this implies that $N \geq \log_5 n$. Let us define $z_i = w_{j_i}$ and $z'_i = \sum_{k=1}^{i-1} z_k$. By the indirect assumption, $\beta(z_i + \frac{1}{m}) \geq z_{i+1} + z_i + z'_i$, thus we have

$$(\beta - 1)z_i - z'_i + 4/m > z_{i+1}. \tag{4}$$

We claim that

$$\frac{z'_{i+1}}{z_{i+1}} > \frac{3}{\beta - 1}\frac{z'_i}{z_i} \tag{5}$$

for every i, or equivalently,

$$(\beta - 1)(z_i^2 + z_i z'_i) - 3z'_i z_{i+1} > 0.$$

By (4), the left hand side is at least

$$(\beta - 1)(z_i^2 + z_i z'_i) - 3z'_i((\beta - 1)z_i - z'_i + 4/m) = (\beta - 1)z_i^2 - 2(\beta - 1)z_i z'_i + 3z'_i(z'_i - 4/m).$$

This is positive if $3(z'_i - 4/m) > (\beta - 1)z'_i$, which holds if m is large enough (e.g. $m > 12/(4 - \beta)$), since $z'_i \geq 1$ and $\beta - 1 < 3$. Since $3/(\beta - 1) > 1$, it follows from (5) that $\frac{z'_N}{z_N} > 3$ for large enough N, and thus

$$\beta w_{j_{N-1}-1} \leq \beta z_N < 4z_N \leq z_N + z'_N,$$

contradicting the indirect assumption. □

By Claim 8, there exists a time point t_i with $\sum_{j:\pi(j)\geq i} w_j > \beta w(J_i)$. By setting $t_{i'} = 0$ (or very close to 0) for $i' < i$ and $t_{i'} = 1$ (or very close to 1) for $i' \geq i$, the schedule can only be a $(\beta - \varepsilon)$-approximation if we do not know the resource arrival times in advance. □

Acknowledgements

The authors are grateful to Erika Bérczi-Kovács and to Matthias Mnich for the helpful discussions.

References

[1] J. Carlier, Problèmes d'ordonnancement à contraintes de ressources: algorithmes et complexité, *Thése, Université Paris VI-Pierre et Marie Curie, Institut de programmation* (1984)

[2] J. Carlier, A.H.G. Rinnooy Kan, Scheduling subject to nonrenewable-resource constraints, *Operations Research Letters* **1** (1982) 52–55

[3] R.L. Graham, E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, Optimization and approximation in deterministic sequencing and scheduling: a survey, *Annals of discrete mathematics* **5** (1979) 287–326

[4] P. Györgyi, T. Kis, Minimizing total weighted completion time on a single machine subject to non-renewable resource constraints, *Journal of Scheduling* **22** (2019) 623–634

[5] P. Györgyi, T. Kis, New complexity and approximability results for minimizing the total weighted completion time on a single machine subject to non-renewable resource constraints, *Discrete Applied Mathematics* **311** (2022) 97-109

[6] T. Kis, Approximability of total weighted completion time with resource consuming jobs, *Operations Research Letters* **43** (2015) 595–598

[7] R. Slowiński, Preemptive scheduling of independent jobs on parallel machines subject to financial constraints, *European Journal of Operational Research* **15** (1984) 366–373

Upper bounds for the necklace folding problems

Endre Csóka[1]

Alfréd Rényi Institute
Budapest, Hungary
csoka.endre@renyi.hu

Zoltán L. Blázsik[2]

Alfréd Rényi Institute
Budapest, Hungary, and
ELKH–ELTE Geometric and Algebraic
Combinatorics Research Group
Budapest, Hungary
blazsik@caesar.elte.hu

Zoltán Király[3]

Department of Computer Science
ELTE Eötvös Loránd University
Budapest, Hungary, and
Alfréd Rényi Institute
Budapest, Hungary
kiraly@cs.elte.hu

Dániel Lenger[4]

Department of Computer Science
ELTE Eötvös Loránd University
Budapest, Hungary, and
Alfréd Rényi Institute
Budapest, Hungary
lengerd@caesar.elte.hu

Abstract: A necklace can be considered as a cyclic list of n red and n blue beads in an arbitrary order. In the necklace folding problem, the goal is to find a large crossing-free matching of pairs of beads of different colors in such a way that there exists a "folding" of the necklace, that is a partition into two contiguous arcs, which splits the beads of any matching edge into different arcs.

We give counterexamples for some conjectures about the necklace folding problem, also known as the separated matching problem. The main conjecture (given independently by three sets of authors) states that $\mu = \frac{2}{3}$, where μ is the ratio of the maximum number of matched beads to the total number of beads.

We refute this conjecture by giving a construction that proves that $\mu \leq 2 - \sqrt{2} < 0.5858 \ll 0.66$. Our construction also applies to the homogeneous model when we match beads of the same color. The full version can be found in [2].

Keywords: separated matching, crossing-free matching, counterexample, construction.

1 Introduction

In the last decades, essentially the same problem known as the necklace folding or the separated matching problem appeared in many areas of mathematics. The problem has two variants which we call the heterogeneous and the homogeneous model. Consider a necklace that consists of $N = 2n$ beads, n red, and n blue ones. In both models, the aim is to find a "folding" of the necklace with a large proper matching of the beads, defined as follows.

[1]The research was supported by Dynasnet European Research Council Synergy project (ERC-2018-SYG 810115).

[2]The research was supported by the Hungarian National Research, Development and Innovation Office, OTKA grant no. SNN 132625.

[3]This research was partially supported by the Hungarian National Research, Development and Innovation Office, OTKA grant no. FK 132524 and by Dynasnet European Research Council Synergy project (ERC-2018-SYG 810115).

A matching M consists of $|M|$ mutually disjoint pairs of beads. In the heterogeneous model, each pair consists of one red and one blue bead while in the homogeneous model, each pair consists of two beads of the same color. The matched pairs will be also called matching edges.

If the beads of the necklace are denoted by $a_1, \ldots, a_N$ in a cyclic order, and $i < j$ are indices, the two *arcs* defined by beads a_i and a_j are contiguous sets of beads: $a_i, a_{i+1}, \ldots, a_j$ and $a_j, \ldots, a_N, a_1, \ldots a_i$. A matching is *crossing-free* if no two matching edges cross each other. (*See Figure 1.*) That is, if the two matching edges are ab and cd, then one arc between a and b is disjoint from the set $\{c, d\}$ while the other arc contains this set entirely.

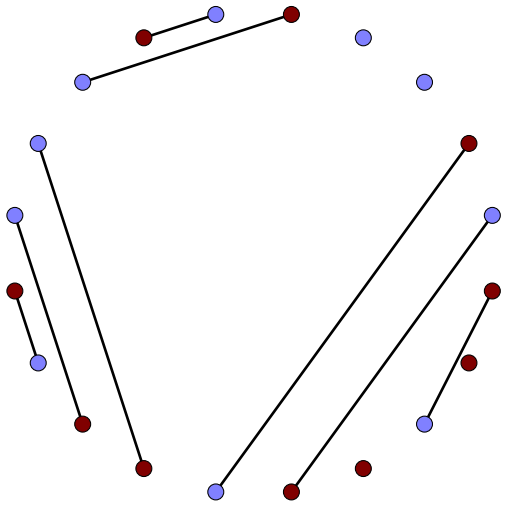


Figure 1: A crossing-free matching in the heterogeneous model

A *secant* partitions the necklace into two arcs, A_1 and A_2. A matching is *secant-respecting* if, for each matching edge, one end is in A_1 while the other end is in A_2. We call a matching *proper* if it is crossing-free and secant-respecting. (*See Figure 2.*)

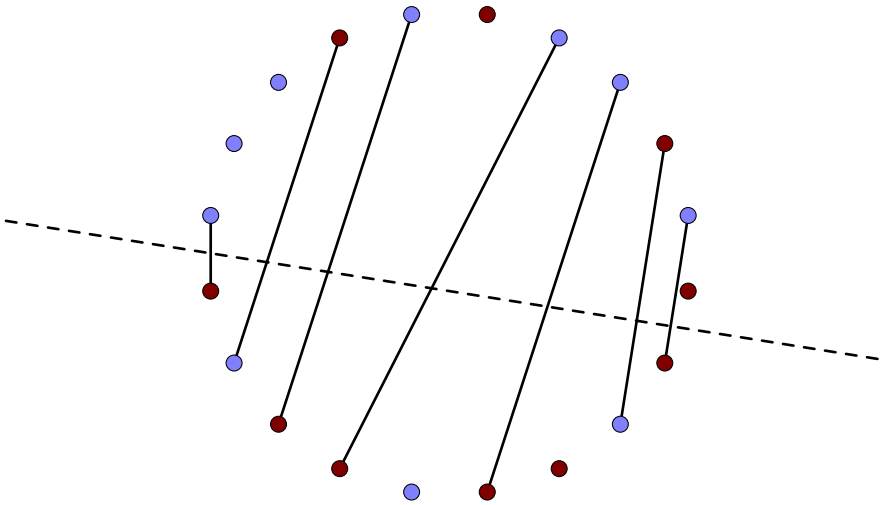


Figure 2: A proper matching (with the respected secant) in the heterogeneous model

Let us remark here that if we drop the secant-respecting condition, then one can easily prove that there is always a crossing-free matching consisting of n edges in the heterogeneous (and $n-1$ edges in

the homogeneous) model.

Proper matchings were called *separated* matchings in [4, 5, 7] where the same problem was considered in a geometric setup. We have n red and n blue points on a circle, the matching edges are considered as segments. A matching is non-crossing if the corresponding segments are pairwise disjoint and a non-crossing matching is separated if there is a straight line that intersects the interior of each of its segments.

Let M be a proper matching. The size $|M|$ of the matching is the number of its edges. A bead is *covered* if it is contained in a matched pair, the number of the covered beads is clearly $2|M|$. Remember that a necklace consists of $N = 2n$ beads, half of them is red, and the other half is blue (i.e., it is balanced). For an even integer N, let $\mathcal{N}(N)$ denote the set of possible balanced necklaces with N beads, and let $\mathcal{M}(L)$ denote the set of proper matchings for a given necklace $L \in \mathcal{N}(N)$ in the heterogeneous model, and $\mu(L) = \max_{M \in \mathcal{M}(L)} 2|M|$, i.e., the maximum number of covered beads in a proper matching. Moreover, let $\mu(N) = \min_{L \in \mathcal{N}(N)} \mu(L)$. Thus $\mu(N)$ is the maximum number of coverable beads in the "worst" necklace. We are interested in $\frac{\mu(N)}{N}$, the ratio of the covered beads to the total number of beads. Remark that it is the same as $|M|/n$ for the maximizing proper matching M. Finally, let $\mu = \limsup_{N \to \infty} \frac{\mu(N)}{N}$. For the homogeneous model, we similarly define $\mu^{\mathrm{hom}}(N)$ and μ^{hom}.

It is trivial that there is a proper matching of size $n/2$ in any given necklace for both models. In the heterogeneous model, one can take an arbitrary secant that cuts the necklace into two arcs each containing n beads. Since the number of blue and red beads are the same, therefore in one of the arcs there are at least as many blue beads as red ones, and in the other arc the opposite is true. Thus we can create a proper matching using the beads of the majority color from each arc. In the homogeneous model, one can take an appropriate secant for which the two arcs have the same number of blue beads. Then there is a proper matching of size $\lfloor n/2 \rfloor$. That is, $\mu \geq \frac{1}{2}$ and $\mu^{\mathrm{hom}} \geq \frac{1}{2}$.

It was very exciting that for 20 years there were no significant improvements about this lower bound, only about the additional $o(1)$ term. However, very recently Mulzer and Valtr [11] managed to improve the lower bound of μ to $(1/2 + \varepsilon)$ for some absolute constant $\varepsilon > 0$.

The story regarding the upper bound is more diversified. Originally only the heterogeneous model was studied. Lyngsø and Pedersen [6] in 1999 proved that $\mu \leq 2/3$, and they conjectured that $\mu = 2/3$. Later independently Kynčl, Pach and Tóth [4, 5] and Brevier, Preissmann and Sebő [1] proved the same upper bound and formulated the same conjecture.

Conjecture 1. *[6, 4, 5, 1] In the heterogeneous model, there is always a proper matching of size at least* $2n/3 - o(n)$*, i.e.,* $\mu = 2/3$*.*

Actually, in [4, 5] a more refined conjecture can be found. For a necklace $L \in \mathcal{N}(N)$, let $\mathrm{mono}(L)$ denote the number of maximal monochromatic arcs, i.e., there is $\mathrm{mono}(L)$ color changes in the necklace, or in other words, the necklace L consist of $\mathrm{mono}(L)/2$ red arcs and $\mathrm{mono}(L)/2$ blue arcs.

Conjecture 2 ([4, 5])**.** *If we restrict ourselves to necklaces* L *where* $\mathrm{mono}(L)/2 = k$*, then for every constant* k *there is always a proper matching of size at least* $\frac{2k-1}{3k-2}n - o(n)$ *in the heterogeneous model.*

However, for the strict connection to Erdős problem (Problem 1) about non-crossing alternating paths, it is enough to assume that $k = o(n)$ (see below). In this case, Conjecture 2 can be read as follows.

Conjecture 3. *In the heterogeneous model, there is always a proper matching of size at least* $2n/3 - o(n)$*, i.e.,* $\mu = 2/3$*, if restrict ourselves for necklaces* $L \in \mathcal{N}(N)$ *where* $\mathrm{mono}(L) = o(n)$*.*

Surprisingly, there are several connections between our problem and some interesting questions from different topics in mathematics. In the sequel, we are going to mention some of these examples as a motivation for our study. The following problem is due to Erdős from the late 80's.

Problem 1. *Determine or estimate the largest number* $\ell = \ell(N)$ *such that, for every set of* $N/2$ *red and* $N/2$ *blue points on a circle, there exists a non-crossing alternating path consisting of* ℓ *vertices.*

Kynčl, Pach and Tóth [4, 5] disproved the original conjecture of Erdős (stating that $\ell(N) = \frac{3}{4}N + o(N)$), and showed the following:

Theorem 4 ([4, 5]). *There exist constants $c, c' > 0$ such that $\frac{1}{2}N + c\sqrt{\frac{N}{\log N}} < \ell(N) < \frac{2}{3}N + c'\sqrt{N}$.*

Moreover, they conjectured that the upper bound is asymptotically tight.

Conjecture 5 ([4, 5]). $|\ell(N) - \frac{2}{3}N| = o(N)$.

Given a necklace $L \in \mathcal{N}(N)$, let $\ell(L)$ denote the maximum length of a non-crossing alternating path. They also proved the following.

Theorem 6 ([4, 5]). $\ell(L) - 2 \cdot \mathrm{mono}(L) - 1 \leq \mu(L) \leq \ell(L)$.

In 2010, Hajnal and Mészáros [3, 7] improved the lower bound on $\ell(N)$ to $N/2 + \Omega(\sqrt{N})$, and also gave a class of configurations reaching the upper bound.

Mészáros [8, 7] investigated separated matchings and found new families of constructions containing at most $\frac{2}{3}N + O(\sqrt{N})$ points in any separated matching. Furthermore, she showed that if the discrepancy is at most three (that is, the difference in the cardinality of the color classes is at most three on any interval), then there are at least $\frac{2}{3}N$ points in the maximum separated matching.

Our main theorem (Theorem 8) disproves Conjecture 5 as well by using Theorem 6.

Interestingly, the above-mentioned problems are closely related to some applied questions about the structure of proteins and some very natural questions about drawing some geometric graphs with non-crossing straight-line edges, too. In 1999, Lyngsø and Pedersen [6] studied folding algorithms in the two-dimensional Hydrophobic-Hydrophilic model (2D HP) for protein structure formation. They provided some approximation algorithms so that the approximation ratio depends exactly on the size of the largest proper matching in our terminology, and conjectured that there always exists a proper matching of size at least $2n/3$.

Moreover, there are some connections between these problems and the investigation of subsequences in circular words over the binary alphabet. One can rephrase Conjecture 1 with this terminology as it states that every binary circular word of length N with an equal number of zeros and ones has an antipalindromic linear subsequence of length at least $2N/3 - o(N)$. Recently, independently from our work, Müllner and Ryzhikov [9, 10] gave a construction (which is essentially the same as our simple construction) that yields an upper bound of $2N/3 + o(N)$ for both the heterogeneous and the homogeneous models (in this latter model we are looking for a palindromic linear subsequence). It seems that they were the first who studied the homogeneous model, and they made the following conjecture.

Conjecture 7 ([9, 10]). $\mu^{\mathrm{hom}} = 2/3$.

We disprove all Conjectures above. Furthermore, we improve the best known upper bound significantly by proving the following theorem.

Theorem 8. *For Construction 1 (see in Section 2), the size of the maximum proper matching is at most $(2 - \sqrt{2})n + o(n)$ in both the heterogeneous and the homogeneous models (i.e., $\mu \leq 2 - \sqrt{2}$ and $\mu^{\mathrm{hom}} \leq 2 - \sqrt{2}$). Moreover, Construction 1 gives an infinite series of necklaces where $\mathrm{mono}(L) = o(n)$.*

Remark 9. *It is not obvious how this theorem disproves Conjecture 2, so we show the transition. By Theorem 8, there exists a specific necklace $L_1 \in \mathcal{N}(N_1)$ with N_1 beads where $\mu(L_1) < 0.6 \cdot N_1$. Let $k = \mathrm{mono}(L_1)/2$. We are giving a counterexample to Conjecture 2 for this k, i.e., an infinite series of necklaces $L_i \in \mathcal{N}(N_i)$ where $\mathrm{mono}(L_i) = 2k$ and $\mu(L_i) < 0.6 \cdot N_i$.*

We construct L_i from L_1 by replacing every bead in L_1 by i consecutive beads of the same color. So $N_i = iN_1$, and obviously $\mathrm{mono}(L_i) = \mathrm{mono}(L_1) = 2k$. Using the fact that in bipartite graphs the weight of the maximum fractional matching is the same as the weight of the maximum matching, it is not hard to prove that $\mu(L_i) = i\mu(L_1)$.

Remark 10. *The problem itself, and also our construction can be defined in a measurable sense, i.e., a necklace is a circle with a measurable two-coloring on its points, and for a proper matching we also require it to be measure-preserving (a red arc with measure λ is matched to a blue arc with measure λ). This is a natural generalization of the discrete problem. Although this language was very useful for finding our counterexample, we present our result in the more classical language of discrete objects. If we used the measurable definition, we may omit the terms $o(N)$ everywhere.*

The full version containing the discussion about the unbalanced case (the number of red beads is between one- and two-thirds of the total number of beads) can be found in [2].

2 Main construction and the proof of Theorem 8

We present here our main construction showing that the size of the maximum proper matching is at most αn where α can be arbitrarily close to $2-\sqrt{2} = 0.5857\ldots < 0.5858$.

Construction 1. *Let $s \geq 2$ be a integer parameter, and let $n = s^{5s+1}$. The necklace consists of s large arcs, each having s^{5s} blue and s^{5s} red beads. Let $L_1, \ldots, L_s$ denote the large arcs.*

L_i is divided into s^{2s-i} red and s^{2s-i} blue arcs, the colors alternates. Let $\ell_{i,j}$ denote the j^{th} arc of L_i, where $1 \leq j \leq 2s^{2s-i}$. The arc $\ell_{i,j}$ always consists of s^{3s+i} beads. In the next step, we will change the color of some beads in each $\ell_{i,j}$ in the following way. Let $\lambda \leq \frac{1}{2}$ be a positive parameter and for a fixed i, let's divide each $\ell_{i,j}$ into s^{s+2i} intervals of size s^{2s-i} and in each of these tiny intervals, change the color of $\lfloor \lambda s^{2s-i} \rfloor$ beads backwards from the clockwise end of the tiny interval. We will refer to those beads whose color were changed as dust in $\ell_{i,j}$. (See Figure 3.)

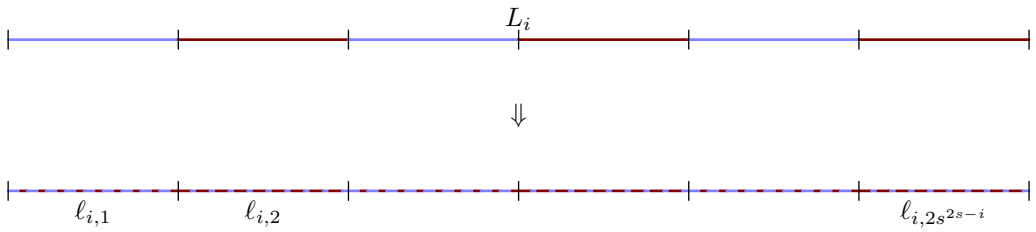


Figure 3: The intervals L_i and $\ell_{i,j}$

First, we bound $\mathrm{mono}(L)$ for a necklace L given by this construction in order to prove the last statement of Theorem 8.

$$\mathrm{mono}(L) = \sum_{i=1}^{s} \sum_{j=1}^{2s^{2s-i}} 2s^{s+2i} = 4s^{3s} \sum_{i=1}^{s} s^i < 4\frac{s^{s+1}-1}{s-1} s^{3s} < 8s^{4s}$$

(as $s \geq 2$), which is $O(n^{4/5}) = o(N)$.

We will see that for $\lambda = 1 - \frac{1}{\sqrt{2}}$, as s tends to ∞, we will get the desired bound, i.e., the upper bound on the size of the proper matching tends to $2-\sqrt{2}$. We will use the little-o notation, e.g., $n/s = o(n)$.

For analyzing this construction we fix an optimal pair of a proper matching and a secant in either the homogeneous or the heterogeneous model, and denote this optimal matching by M. The secant may split at most two large arcs, call them L_p and L_r. If, e.g., one end of the secant is between the large arcs L_j and L_{j+1}, then let $p = j$. We may assume that $p < r$ (if $p = r$, then every matching edge has one end in L_p, so $|M| \leq n/s = o(n)$).

Let $M' \subseteq M$ consist of those matching edges for which no end-vertex is inside the set $L_p \cup L_r$. Obviously, $|M| \leq |M'| + 2n/s = |M'| + o(n)$. We call a pair of indices (g, h) *bonded* if there exists at least one edge of M' connecting L_g and L_h. (*See Figure 4.*)

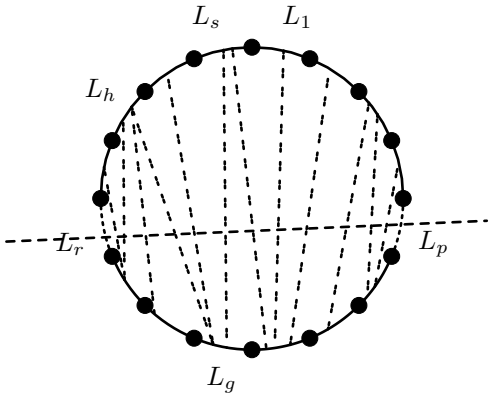


Figure 4: Bonded pairs.

Lemma 11. *The number of the bonded pairs is at most* $s-3$.

Proof. Consider the auxiliary graph with vertices $\{1,2,\dots s\}\setminus\{p,r\}$, and connect two vertices if the corresponding pair is bonded. There is no cycle in this graph, otherwise a cycle yields a crossing in M. Thus the auxiliary graph may have at most $(s-2)-1$ edges. □

Let I be an interval. If x and y are the first and last M'-matched bead in I, and $M'(x)$ and $M'(y)$ are their matched partners, then we assign the arc spanned by $M'(x)$ and $M'(y)$ to I, denote this arc by $M'(I)$.

Let (g,h) be a bonded pair, where $g<h$. For $i\in\{1,\dots,s^{2s-g}\}$ let's call a pair of intervals $(\ell_{g,2i-1},\ell_{g,2i})$ (g,h)-*regular*, if there exists $j\in\{1,\dots 2s^{2s-h}\}$ such that $M'(\ell_{g,2i-1}\cup\ell_{g,2i})\subseteq\ell_{h,j}$. Let's call an edge of M' *regular* if one of the end-vertices is in a (g,h)-regular pair for some $g<h$. Denote the set of regular edges by M''. An edge of M' is called a (g,h)-edge if its end-vertices are in L_g and in L_h, respectively; and irregular if it is not regular. (*See Figure 5.*)

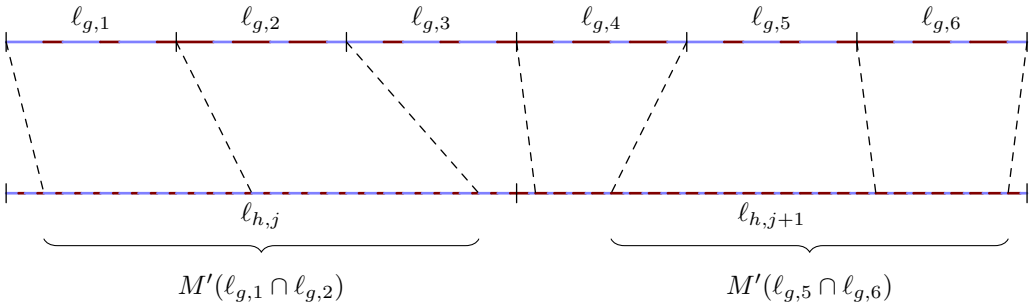


Figure 5: $(\ell_{g,1},\ell_{g,2})$ and $(\ell_{g,5},\ell_{g,6})$ are (g,h)-regular, because $M'(\ell_{g,1}\cap\ell_{g,2})\subseteq\ell_{h,j}$ and $M'(\ell_{g,5}\cap\ell_{g,6})\subseteq\ell_{h,j+1}$, but $(\ell_{g,3},\ell_{g,4})$ is not (g,h)-regular.

Lemma 12. $|M'|\le|M''|+6n/s=|M''|+o(n)$.

Proof. Consider a bonded pair (g,h) for some $g<h$. First, we are going to bound the number of irregular (g,h)-edges.

Take an $i\in\{1,\dots,s^{2s-g}\}$ for which $M'(\ell_{g,2i-1}\cup\ell_{g,2i})\cap L_h\neq\emptyset$ but $(\ell_{g,2i-1},\ell_{g,2i})$ is not (g,h)-regular. It means that either there exist a "bad index" $j\in\{1,\dots,2s^{2s-h}-1\}$ such that both $M'(\ell_{g,2i-1}\cup\ell_{g,2i})\cap\ell_{h,j}$ and $M'(\ell_{g,2i-1}\cup\ell_{g,2i})\cap\ell_{h,j+1}$ are non-empty. We also call $j=0$ bad if $M'(\ell_{g,2i-1}\cup\ell_{g,2i})\cap L_{h-1}$ is

non-empty (where $L_0 = L_s$), and $j = 2s^{2s-h}$ bad if $M'(\ell_{g,2i-1} \cup \ell_{g,2i}) \cap L_{h+1}$ is non-empty (where $L_{s+1} = L_1$).

Moreover, any j can be a bad index at most once therefore the number of such i's is at most $2s^{2s-h}+1$. Even if all the beads in these non-(g,h)-regular pairs are M'-covered, we got rid of at most $(2s^{2s-h}+1) \cdot 2|\ell_{g,2i}| = (2s^{2s-h}+1) \cdot (2s^{3s+g}) < 6s^{5s-(h-g)} \leq 6s^{5s-1}$, since $h > g$. This is true for any bonded pairs, hence altogether we lost at most $(s-3)6s^{5s-1} < 6s^{5s} = 6n/s = o(n)$ M'-edges. □

From now on, we estimate the number of regular edges. For the sake of simplicity, we will omit the floor and ceiling functions, because the difference in the result is again $o(n)$.

For $g < h$, let's fix a bonded pair (g,h). Consider a (g,h)-regular pair of intervals $(\ell_{g,2i-1}, \ell_{g,2i})$ such that $M'(\ell_{g,2i-1} \cup \ell_{g,2i}) \subseteq \ell_{h,j}$.

Until this point, there was no difference between the homogeneous and the heterogeneous case. In the sequel, there still will not be any significant difference, the calculations work the same way in both cases. We now present the calculation for the homogeneous case, and we will assume that the "main" color of $\ell_{h,j}$ is blue (i.e. the dust is red), the "main" color of $\ell_{g,2i-1}$ is red, thus the "main" color of $\ell_{g,2i}$ is blue.

Let's denote the *efficiency* of the matching M'' on an interval I with

$$\mathrm{eff}(I) = \frac{\#\text{ of beads in } I \cup M'(I) \text{ covered by } M''}{|I \cup M'(I)|}.$$

In the following lemma, we will show that the efficiency cannot exceed $2 - \sqrt{2} + o(1)$ for a suitable λ.

Lemma 13. $\mathrm{eff}(\ell_{g,2i-1} \cup \ell_{g,2i}) \leq 2 - \sqrt{2} + o(1)$ *if* $\lambda = 1 - \frac{1}{\sqrt{2}}$.

Proof. Recall that $\ell_{g,2i-1}$ (and also $\ell_{g,2i}$) is divided into s^{s+2g} red and s^{s+2g} blue monochromatic intervals. We will call them $\ell_{g,2i-1}^{(\mathrm{red},1)}, \dots \ell_{g,2i-1}^{(\mathrm{red},s^{s+2g})}, \ell_{g,2i-1}^{(\mathrm{blue},1)}, \dots \ell_{g,2i-1}^{(\mathrm{blue},s^{s+2g})}$, where the *red* and *blue* indicates the color of the interval.

Also recall that, $\ell_{h,j}$ is divided into blue and red monochromatic intervals (the color alternates) whose sizes are $(1-\lambda)s^{2s-h}$ and λs^{2s-h}, respectively. We define numbers a_k, b_k, c_k and d_k for $1 \leq k \leq s^{s+2g}$ in the following way.

Assume that the number of M''-covered beads from $\ell_{g,2i-1}^{(\mathrm{red},k)}$ is x. Then $a_k = \frac{x}{\lambda s^{2s-h}}$, i.e., the necessary number of small red intervals (dust) from $\ell_{h,j}$ to cover that many beads.

Similarly, assume that the number of M''-covered beads from $\ell_{g,2i-1}^{(\mathrm{blue},k)}$ is x. Then $b_k = \frac{x}{(1-\lambda)s^{2s-h}}$, i.e., the necessary number of blue intervals from $\ell_{h,j}$ to cover that many beads.

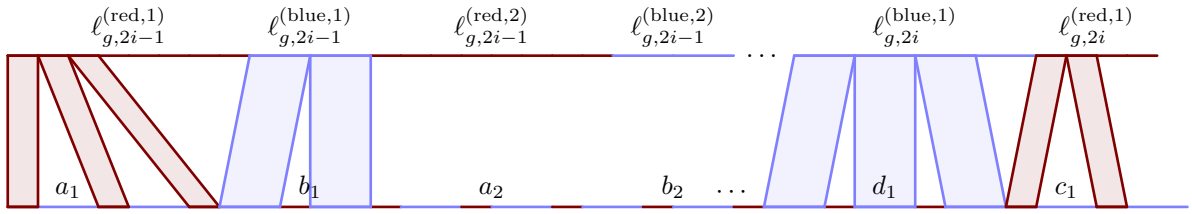


Figure 6: The definition of a_k, b_k, c_k and d_k in the homogeneous case

We define c_k and d_k in the same way for $\ell_{g,2i}^{(\mathrm{red},k)}$ and $\ell_{g,2i}^{(\mathrm{blue},k)}$, respectively. (*See Figure 6.*) It is easy to see that $a_k \leq \frac{1-\lambda}{\lambda} s^{h-g}$, $b_k \leq \frac{\lambda}{1-\lambda} s^{h-g}$, and $c_k, d_k \leq s^{h-g}$.

The number of M''-covered beads in $(\ell_{g,2i-1} \cup \ell_{g,2i}) \cup M'(\ell_{g,2i-1} \cup \ell_{g,2i})$ is

$$2\left(\sum a_k \lambda s^{2s-h} + \sum b_k (1-\lambda) s^{2s-h} + \sum c_k \lambda s^{2s-h} + \sum d_k (1-\lambda) s^{2s-h}\right).$$

In all of the sums, k runs from 1 up to s^{s+2g}, so we use the following shorthands. Let

$$A = \sum_{k=1}^{s^{s+2g}} a_k,\ B = \sum_{k=1}^{s^{s+2g}} b_k,\ C = \sum_{k=1}^{s^{s+2g}} c_k,\ D = \sum_{k=1}^{s^{s+2g}} d_k.$$

Obviously, $|\ell_{g,2i-1} \cup \ell_{g,2i}| = 2s^{3s+g}$. We will give a lower bound on $|M'(\ell_{g,2i-1} \cup \ell_{g,2i})|$. To cover the M''-matched beads in $\ell_{g,2i-1}^{(\mathrm{red},k)}$, we need at least a_k monochromatic red interval from $\ell_{h,j}$, thus at least $a_k - 1$ monochromatic blue interval remained unused, so $M'(\ell_{g,2i-1}^{(\mathrm{red},k)}) \geq (a_k - 1)s^{2s-h}$. Similarly $M'(\ell_{g,2i-1}^{(\mathrm{blue},k)}) \geq (b_k - 1)s^{2s-h}$, $M'(\ell_{g,2i}^{(\mathrm{red},k)}) \geq (c_k - 1)s^{2s-h}$ and $M'(\ell_{g,2i}^{(\mathrm{blue},k)}) \geq (d_k - 1)s^{2s-h}$.

Altogether, we get that $\mathrm{eff}(\ell_{g,2i-1} \cup \ell_{g,2i})$ is at most

$$2\frac{\lambda A s^{2s-h} + (1-\lambda)Bs^{2s-h} + \lambda C s^{2s-h} + (1-\lambda)Ds^{2s-h}}{[\sum(a_k-1)s^{2s-h} + \sum(b_k-1)s^{2s-h} + \sum(c_k-1)s^{2s-h} + \sum(d_k-1)s^{2s-h}] + 2s^{3s+g}} =$$

$$= 2\frac{\lambda A + (1-\lambda)B + \lambda C + (1-\lambda)D}{[A+B+C+D-4s^{s+2g}] + 2s^{3s+g}/s^{2s-h}} =$$

$$2\frac{\lambda A + (1-\lambda)B + \lambda C + (1-\lambda)D}{A+B+C+D-4s^{s+2g}+2s^{s+g+h}}.$$

Let's denote this last expression by $\mathrm{eff}\,(A,B,C,D)$. First, we will show that this expression is monotone increasing in B and D.

$$\mathrm{eff}\,(A,B,C,D) = 2(1-\lambda) + 2\frac{(2\lambda-1)(A+C) - (1-\lambda)(2s^{s+g+h} - 4s^{s+2g})}{A+B+C+D+(2s^{s+g+h}-4s^{s+2g})}.$$

As $\lambda \leq \frac{1}{2}$ and $s^{s+g+h} \geq s^{s+2g+1} \geq 2s^{s+2g}$, we have that $2\lambda - 1 \leq 0$ and $2s^{s+g+h} - 4s^{s+2g} > 0$, so the numerator of the second term is negative. Thus we can increase the value of this expression by choosing B and D as large as possible which yields:

$$\mathrm{eff}\,(A,B,C,D) \leq \mathrm{eff}\left(A, \frac{\lambda}{1-\lambda}s^{s+g+h}, C, s^{s+g+h}\right).$$

We will do the same trick for A and C.

$$\mathrm{eff}\left(A, \frac{\lambda}{1-\lambda}s^{s+g+h}, C, s^{s+g+h}\right) =$$

$$= 2\frac{\lambda A + (1-\lambda)\frac{\lambda}{1-\lambda}s^{s+g+h} + \lambda C + (1-\lambda)s^{s+g+h}}{A + \frac{\lambda}{1-\lambda}s^{s+g+h} + C + s^{s+g+h} + 2s^{s+g+h} - 4s^{s+2g}} =$$

$$= 2\frac{\lambda A + \lambda C + s^{s+g+h}}{A + C + (\frac{\lambda}{1-\lambda}+3)s^{s+g+h} - 4s^{s+2g}} =$$

$$= 2\lambda + 2\frac{\left[1 - \lambda(\frac{\lambda}{1-\lambda}+3)\right]s^{s+g+h} + 4\lambda s^{s+2g}}{A + C + \left(\frac{\lambda}{1-\lambda}+3\right)s^{s+g+h} - 4s^{s+2g}} =$$

$$= 2\lambda + 2\frac{\frac{2\lambda^2-4\lambda+1}{1-\lambda}s^{s+g+h} + 4\lambda s^{s+2g}}{A + C + \left(\frac{\lambda}{1-\lambda}+3\right)s^{s+g+h} - 4s^{s+2g}}.$$

If $\lambda = 1 - \frac{1}{\sqrt{2}}$, then $2\lambda^2 - 4\lambda + 1 = 0$, so

$$\mathrm{eff}\left(A, \frac{\lambda}{1-\lambda}s^{s+g+h}, C, s^{s+g+h}\right) =$$

$$= 2\lambda + 2\frac{4\lambda s^{s+2g}}{A + C + \left(\frac{\lambda}{1-\lambda} + 3\right) s^{s+g+h} - 4s^{s+2g}}.$$

The numerator of the second term is $\Theta(s^{s+2g})$ while the denominator is $\Theta(s^{s+g+h})$ because $A \leq \frac{1-\lambda}{\lambda} s^{s+g+h}$ and $C \leq s^{s+g+h}$. Thus

$$\text{eff}\left(A, \frac{\lambda}{1-\lambda} s^{s+g+h}, C, s^{s+g+h}\right) = 2\lambda + O(s^{g-h}) = 2 - \sqrt{2} + o(1). \qquad \square$$

We have already proved that the number of those edges in the matching M which are not in M'' is negligible. We can partition the rest of the edges (the regular ones) into disjoint subsets according to the (a, b)-bonded pairs determined by their beads. In every such (a, b)-bonded pair (for some $a < b$), we can repeat the argument of Lemma 13. Hence, we can conclude that in this construction the size of a proper matching is at most $\left(2 - \sqrt{2} + o(1)\right) n \approx 0.5858n$. $\square$

Remark 14. *In the heterogeneous case, we assume that $\ell_{h,j}$ is divided into red and blue monochromatic intervals (the color alternates) whose sizes are $(1-\lambda)s^{2s-h}$ and λs^{2s-h}, respectively. We define numbers a_k, b_k, c_k and d_k for $1 \leq k \leq s^{s+2g}$ in the following way. (*See Figure 7.*)*

Let a_k denote the necessary number of small blue intervals (dust) from $\ell_{h,j}$ to cover the M''-covered beads from $\ell_{g,2i-1}^{(\text{red},k)}$ beads. Similarly, let b_k denote the necessary number of red intervals from $\ell_{h,j}$ to cover the M''-covered beads from $\ell_{g,2i-1}^{(\text{blue},k)}$ beads. We define c_k and d_k in the same way for $\ell_{g,2i}^{(\text{red},k)}$ and $\ell_{g,2i}^{(\text{blue},k)}$, respectively. It is easy to see that $a_k \leq \frac{1-\lambda}{\lambda} s^{h-g}$, $b_k \leq \frac{\lambda}{1-\lambda} s^{h-g}$, and $c_k, d_k \leq s^{h-g}$.

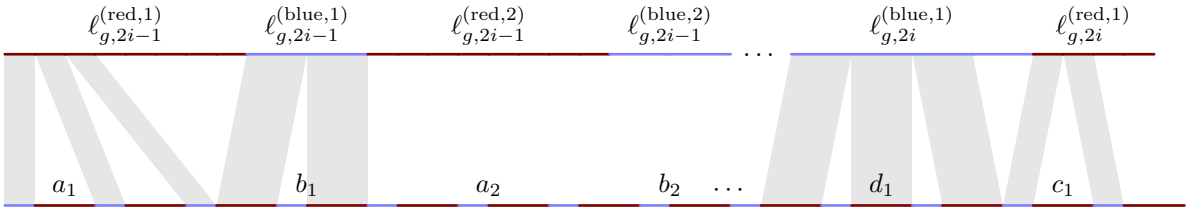


Figure 7: The definition of a_k, b_k, c_k and d_k in the heterogeneous case

The number of M''-covered beads in $(\ell_{g,2i-1} \cup \ell_{g,2i}) \cup M'(\ell_{g,2i-1} \cup \ell_{g,2i})$ is

$$2\left(\sum a_k \lambda s^{2s-h} + \sum b_k (1-\lambda) s^{2s-h} + \sum c_k \lambda s^{2s-h} + \sum d_k (1-\lambda) s^{2s-h}\right).$$

From this point on the same calculation can give us the proof of Lemma 13 in the heterogeneous case and then conclude that $\mu \leq 2 - \sqrt{2}$, too.

Acknowledgement

This research was started during the 9th Emléktábla Workshop, 2019. The authors are thankful to the organizers for inviting them. We also thank D. Pálvölgyi, G. Damásdi, T. Fleiner and Zs. Jankó for valuable questions and observations.

References

[1] G. Brevier, M. Preissmann and A. Sebő, personal communication (2004).

[2] E. Csóka, Z. L. Blázsik, Z. Király and D. Lenger, Upper bounds for the necklace folding problems, *Journal of Combinatorial Theory, Series B*, **157**, (2022), pp. 123–143.

[3] P. Hajnal and V. Mészáros, A note on noncrossing path in colored convex sets, manuscript, (2010).

[4] J. Kynčl, J. Pach and G. Tóth, Long alternating paths in bicolored point sets, in Graph Drawing (J. Pach, ed.), Lecture Notes in Computer Science **3383**, Springer-Verlag, Berlin, (2004), pp. 340–348.

[5] J. Kynčl, J. Pach and G. Tóth, Long alternating paths in bicolored point sets, *Discrete Mathematics*, **308**, (2008), pp. 4315–4322.

[6] R. B. Lyngsø and C. N. S. Pedersen, Protein Folding in the 2D HP Model, *BRICS Report Series*, **RS-99-16**, (1999).

[7] V. Mészáros, Extremal problems on planar point sets, *Ph.D. thesis*, doktori.bibl.u-szeged.hu/688/1/mvdoktori.pdf, (2011).

[8] V. Mészáros, Separated matchings and small discrepancy colorings. *Computational Geometry*, *Lecture Notes in Comput. Sci.*, **7579**, Springer, Cham, (2011), pp. 236–248.

[9] C. Müllner and A. Ryzhikov, Palindromic Subsequences in Finite Words. *arXiv*, **1901.07502**, (2019).

[10] C. Müllner and A. Ryzhikov, Palindromic subsequences in finite words. In Proc 13th Int. Conf. Language and Automata Theory and Applications (LATA), (2019), pp. 460–468.

[11] W. Mulzer and P. Valtr, Long alternating paths exist. *arXiv*, **2003.13291**, (2020).

Reconfiguration of Graph Orientations with Connectivity Constraints[1]

Takehiro Ito
Tohoku University
takehiro@tohoku.ac.jp

Yuni Iwamasa
Kyoto University
iwamasa@i.kyoto-u.ac.jp

Naonori Kakimura
Keio University
kakimura@math.keio.ac.jp

Naoyuki Kamiyama
Kyushu University
kamiyama@imi.kyushu-u.ac.jp

Yusuke Kobayashi
Kyoto University
yusuke@kurims.kyoto-u.ac.jp

Shun-ichi Maezawa
Tokyo University of Science
maezawa.mw@gmail.com

Yuta Nozaki
Hiroshima University
nozakiy@hiroshima-u.ac.jp

Yoshio Okamoto
The University of Electro-Communications
okamotoy@uec.ac.jp

Kenta Ozeki
Yokohama National University
ozeki-kenta-xr@ynu.ac.jp

Abstract: Graph orientation is the process of orienting the edges of an undirected graph to obtain a directed graph, and is a topic that has been actively studied in the fields of combinatorial optimization and graph theory. In this paper, we present our results on "reconfiguration of graph orientations", where the directions of some edges are flipped one by one while maintaining a certain connectivity constraint. Our main result is the following: for an arbitrary orientation of a $2k$-edge-connected undirected graph, we can monotonically increase the edge-connectivity by flipping the directions of some edges one by one, and finally obtain a k-edge-connected orientation. This result strengthens the classical Nash-Williams' theorem, and is useful when discussing the connectivity of the edge-flip graph of k-edge-connected orientations.

Graph connectivity, Graph orientation, Combinatorial reconfiguration, Nash-Williams' Theorem, Edge-flip graph

1 Increasing Edge-Connectivity by Edge-Flips

For an undirected graph $G = (V, E)$ with possible multiple edges, an *orientation* of G is a directed graph $D = (V, A)$ obtained from G by replacing each undirected edge $\{u, v\} \in E$ with a directed edge $(u, v) \in A$ or $(v, u) \in A$. An old result by Robbins [6] states that an undirected graph G has a strongly connected orientation if and only if G is 2-edge-connected. Robbins' theorem was extended by Nash-Williams [5] stating that an undirected graph G has a k-edge-connected orientation if and only if G is $2k$-edge-connected.

In this paper, we consider the reorientation of directed graphs, where the directions of some edges are flipped one by one while maintaining a certain connectivity constraint. This has practical importance since simultaneous edge flips can be difficult to implement or control in some real-world situations such as traffic management, and the reduction of edge-connectivity in intermediate orientations may cause the

[1]The full version of this paper is available at [4]. Research is supported by JSPS KAKENHI Grant Numbers JP20H05793, JP20H05795, JP20K11670, JP20K11692, JP19K11814, JP18H04091, JP18H05291, and JP21H03397, Japan.

loss of network quality. To make the discussion more precise, we define an *edge flip* (or a *flip* for short) of a directed edge (u, v) as an operation that replaces (u, v) by (v, u), i.e., reverses the direction of (u, v). For directed graphs D and D', we denote $D \to D'$ if D' is obtained from D by a single edge flip.

Our main contribution is to show that for any orientation of a $2k$-edge-connected undirected graph G, there exists a sequence of edge flips such that the orientations of G obtained by the successive edge flips have non-decreasing edge-connectivity and the resulting orientation is k-edge-connected. Here, we recall that the *edge-connectivity* of a directed graph $D = (V, A)$ is the maximum integer λ such that every non-empty subset $X \subsetneq V$ has at least λ edges leaving X, and is denoted by $\lambda(D)$. Formally, our main result is stated as follows.

Theorem 1 *Let k be a non-negative integer. Let $G = (V, E)$ be an undirected $2k$-edge-connected graph and $D = (V, A)$ be an orientation of G with $\lambda(D) \leq k$. Then, there exist orientations $D_1, D_2, \ldots, D_\ell$ of G such that $\ell \leq (k - \lambda(D))|V|^3$, $D \to D_1 \to D_2 \to \cdots \to D_\ell$, and $\lambda(D) \leq \lambda(D_1) \leq \lambda(D_2) \leq \cdots \leq \lambda(D_\ell) = k$. Furthermore, such orientations $D_1, \ldots, D_\ell$ can be found in polynomial time.*

It is not difficult see that Theorem 1 is obtained by applying the following theorem repeatedly.

Theorem 2 *Let k be a non-negative integer. Let $G = (V, E)$ be an undirected $(2k + 2)$-edge-connected graph and $D = (V, A)$ be a k-edge-connected orientation of G. Then, there exist orientations $D_1, D_2, \ldots, D_\ell$ of G such that $\ell \leq |V|^3$, $D \to D_1 \to D_2 \to \cdots \to D_\ell$, $\lambda(D_i) \geq k$ for $i \in \{1, \ldots, \ell - 1\}$, and $\lambda(D_\ell) \geq k+1$. Furthermore, such $D_1, \ldots, D_\ell$ can be found in polynomial time.*

Thus, in order to obtain Theorem 1, it suffices to show Theorem 2. We here briefly explain the outline of the proof of Theorem 2; see [4] for the full proof.

For $D = (V, A)$, we fix a vertex $r \in V$ arbitrarily. Define $\mathcal{F}_{\mathrm{out}}(D)$ and $\mathcal{F}_{\mathrm{in}}(D)$ as

$$\mathcal{F}_{\mathrm{out}}(D) := \{X \subseteq V - r \mid \delta^+_D(X) = k\} \cup \{V\},$$
$$\mathcal{F}_{\mathrm{in}}(D) := \{X \subseteq V - r \mid \delta^-_D(X) = k\} \cup \{V\}.$$

We can easily see that D is $(k + 1)$-edge-connected if and only if $\mathcal{F}_{\mathrm{out}}(D) = \mathcal{F}_{\mathrm{in}}(D) = \{V\}$. Define $\mathcal{F}_{\mathrm{min}}(D)$ as the set of all inclusionwise minimal sets in $\mathcal{F}_{\mathrm{out}}(D) \cup \mathcal{F}_{\mathrm{in}}(D)$. Actually, $\mathcal{F}_{\mathrm{min}}(D)$ consists of disjoint sets (see [4]).

In our proof of Theorem 2, by flipping some edges in D, we decrease the value of

$$\mathsf{val}(D) := \sum_{X \in \mathcal{F}_{\mathrm{min}}(D)} (|V| - |X|).$$

Indeed, we show that we can decrease $\mathsf{val}(D)$ by applying at most $|V|$ edge flips. We repeat this procedure as long as $\mathsf{val}(D)$ is positive. If this value becomes 0, then $\mathcal{F}_{\mathrm{min}} = \{V\}$. This means that $\mathcal{F}_{\mathrm{out}} = \mathcal{F}_{\mathrm{in}} = \{V\}$, and hence D is $(k + 1)$-edge-connected. Note that we decrease the value of $\mathsf{val}(D)$ at most $|V|^2$ times, because $\mathsf{val}(D)$ is integral and $\mathsf{val}(D) \leq |V|^2$. Therefore, the total number of edge flips is at most $|V|^3$.

2 Connectedness of the Edge-flip Graph

As a consequence of Theorem 1, we obtain a result on the connectedness of the edge-flip graph of k-edge-connected orientations. For an undirected graph $G = (V, E)$, we define the *edge-flip graph* $\mathcal{G}_k(G)$ to be the graph whose vertices correspond to the k-edge-connected orientations of G, and two orientations are joined by an edge in the edge-flip graph if and only if one is obtained from the other by a single edge flip. We consider the question that asks when $\mathcal{G}_k(G)$ is connected.

When $k = 1$, this question is completely answered. Greene and Zaslavsky [3] proved by hyperplane arrangements that the edge-flip graph $\mathcal{G}_1(G)$ is connected if and only if G is 3-edge-connected. Fukuda, Prodon, and Sakuma [2] gave a graph-theoretic proof for the same fact. As a higher-edge-connectedness analogue of this fact, we give a partial answer to this question for $k \geq 2$.

Theorem 3 *Let $k \geq 1$. If G is $(2k+2)$-edge-connected, then the edge-flip graph $\mathcal{G}_k(G)$ is connected.*

Theorem 3 is obtained as a corollary of Theorem 1, combined with the following theorem.

Theorem 4 (Frank [1]) *Let $k \geq 1$ be an integer, $G = (V, E)$ be a $2k$-edge-connected undirected graph, and D_1, D_2 be two k-edge-connected orientations of G. Then, D_1 and D_2 can be transformed with each other by a sequence of path/cycle flips in such a way that all the intermediate orientations are k-edge-connected.*

Here, a *path/cycle flip* is an operation that flips all the edges of a directed path or a directed cycle simultaneously.

To prove Theorem 3, for k-edge-connected orientations D_1 and D_2 of G, we show that D_1 can be transformed to D_2 by a sequence of edge flips, while maintaining the k-edge-connectedness. Below is our strategy to transform D_1 to D_2; see the full paper [4] for the complete proof.

1. We apply Theorem 1 to transform D_1 to a $(k+1)$-edge-connected orientation D_1' by edge flips so that all the intermediate orientations are k-edge-connected. This can be done by the assumption that G is $(2k+2)$-edge-connected. We apply the same procedure to D_2 to obtain a $(k+1)$-edge-connected orientation D_2'.

2. We next apply Theorem 4 to transform D_1' to D_2'. Since operations in Theorem 4 are path/cycle flips, we need to turn them into sequences of edge flips. We emphasize that all the intermediate orientations will be k-edge-connected, but not necessarily $(k+1)$-edge-connected.

3. Finally, we consider the reverse sequence of edge flips that transformed D_2 to D_2' from the first step. Combining them, we obtain a sequence of edge flips that transforms D_1 to D_2 such that all the intermediate orientations are k-edge-connected.

We do not know if the $(2k+2)$-edge-connectedness can be replaced with the $(2k+1)$-edge-connectedness when $k \geq 2$. However, we know that we cannot replace it with the $2k$-edge-connectedness. Indeed, there exists a $2k$-edge-connected graph G such that $\mathcal{G}_k(G)$ is disconnected even when $k = 1$ (e.g. consider the clockwise orientation and the counterclockwise orientation of a 3-cycle). Note that if the edge-connectivity of G is less than $2k$, then $\mathcal{G}_k(G)$ is not defined (or it is the null graph with no vertices).

References

[1] A. Frank, A note on k-strongly connected orientations of an undirected graph, *Discret. Math. 39*, 1 (1982), 103–104.

[2] K. Fukuda, A. Prodon, and T. Sakuma, Notes on acyclic orientations and the shelling lemma, *Theor. Comput. Sci. 263*, 1-2 (2001), 9–16.

[3] C. Greene, and T. Zaslavsky, On the interpretation of Whitney numbers through arrangements of hyperplanes, zonotopes, non-Radon partitions, and orientations of graphs, *Trans. Amer. Math. Soc. 280* (1983), 97–126.

[4] T. Ito, Y. Iwamasa, N. Kakimura, N. Kamiyama, Y. Kobayashi, S. Maezawa, Y. Nozaki, Y. Okamoto, and K. Ozeki: Monotone edge flips to an orientation of maximum edge-connectivity à la Nash-Williams, *ACM Transactions on Algorithms*, to appear. The preprint is available at arXiv:2110.11585.

[5] C. S. J. A. Nash-Williams, On orientations, connectivity and odd-vertex-pairings in finite graphs, *Canadian Journal of Mathematics 12* (1960), 555–567.

[6] H. E Robbins, A theorem on graphs, with an application to a problem of traffic control, *The American Mathematical Monthly 46*, 5 (1939), 281–283.

Lipschitz Continuous Graph Algorithms

Soh Kumabe[1]

The University of Tokyo
7-3-1, Hongo, Bunkyo-ward, Tokyo, Japan
soh_kumabe@mist.i.u-tokyo.ac.jp

Yuichi Yoshida[2]

National Institute of Informatics
2-1-2, Hitotsubashi, Chiyoda ward, Tokyo, Japan
yyoshida@nii.ac.jp

Abstract: Adversarial attacks, in which small perturbations to the input can cause a large change in the prediction of a trained model, have been widely observed in the machine learning community. As graph algorithms are commonly used for decision-making and knowledge discovery, it is important to design robust algorithms against such attacks. In this study, we investigate the Lipschitz continuity of algorithms for (weighted) graph problems as a measure of robustness against adversarial attacks. Our study initiates a systematic study of the Lipschitz continuity of algorithms for graph problems.

In this study, we focus on a specific notion of Lipschitz continuity that is invariant under scaling of weights. Using this measure, we provide Lipschitz continuous algorithms and lower bounds for the minimum spanning tree problem, the shortest path problem, and the maximum weight matching problem.

We also consider another Lipschitz continuity notion induced by a natural mapping that maps the output solution to its characteristic vector. We show that no Lipschitz continuous algorithms exist for this Lipschitz notion. Instead, we design algorithms with bounded pointwise Lipschitz constants for the minimum spanning tree problem and the maximum weight bipartite matching problem.

Keywords: Lipschitz continuity, Sensitivity, Graph algorithms, Approximation algorithms

1 Introduction

1.1 Backgrounds

In the field of machine learning, adversarial attacks are small perturbations to the input that can cause a large change in the prediction of a trained model [6, 14]. These attacks pose a threat to the security of machine learning-based systems, and there is ongoing research on training models that are robust against them [2, 10]. Surveys on this topic can be found in [1, 19, 23].

While there has been significant attention on adversarial attacks in the context of machine learning, there has been little research on designing graph algorithms that are robust against such attacks. In this study, we consider the *Lipschitz continuity* of graph algorithms as a measure of robustness against adversarial attacks, and we provide algorithms with small *Lipschitz constants* for various graph problems. In this paper, we design graph algorithms that are robust against adversarial attacks, and we provide a framework for systematically designing such algorithms.

Before presenting our algorithmic results, we discuss how the Lipschitz continuity of a graph algorithm must be defined.

[1]Research is supported by JST, PRESTO Grant Number JPMJPR192B.
[2]Research is supported by JST, PRESTO Grant Number JPMJPR192B.

1.2 Lipschitz Continuity

Metrics. Let $\mathcal{A}$ be an algorithm that, given a graph $G = (V, E)$ and a weight vector $w \in \mathbb{R}^E_{\geq 0}$ over edges, outputs an edge set $\mathcal{A}(G, w) \subseteq E$. Then, how should we define the Lipschitz constant of $\mathcal{A}$, or more specifically, which metric should we impose on the input and output spaces?

In this study, we always adopt the ℓ_1 metric for the input space, that is, the distance between two weight vectors $w, w' \in \mathbb{R}^E_{\geq 0}$ is defined to be $\|w - w'\|_1$. We do so because for combinatorial problems, it is natural to assume that the distance between w and w' is calculated as the sum of the distances between $w(e)$ and $w'(e)$ over $e \in E$, and the ℓ_1 metric satisfies this property.

We also use the ℓ_1 metric for the output space. Depending on how we map the output edge set to a vector in the ℓ_1 space, we can think of the following two variations.

(Unweighted mapping) We map an edge set $F \subseteq E$ to the characteristic vector $\mathbf{1}_F \in \mathbb{R}^E$ of F, where $\mathbf{1}_F(e) = 1$ if $e \in F$ and 0 otherwise. Then for two edge sets $F, F' \subseteq E$, we define

$$d_u(F, F') := \|\mathbf{1}_F - \mathbf{1}_{F'}\|_1 = |F \triangle F'|.$$

(Weighted mapping) We map an edge set $F \subseteq E$ to a vector $\sum_{e \in F} w(e)\mathbf{1}_e$ using the weight vector $w \in \mathbb{R}^E_{\geq 0}$, where $\mathbf{1}_e \in \mathbb{R}^E$ is the characteristic vector of $e \in E$, that is, $\mathbf{1}_e(f) = 1$ if $f = e$ and 0 otherwise. Then for two edge sets $F, F' \subseteq E$ and weight vectors $w, w' \in \mathbb{R}^E_{\geq 0}$, we define

$$\begin{aligned} d_w((F, w), (F', w')) &:= \left\| \sum_{e \in F} w(e)\mathbf{1}_e - \sum_{e \in F'} w'(e)\mathbf{1}_e \right\|_1 \\ &= \sum_{e \in F \cap F'} |w(e) - w'(e)| + \sum_{e \in F \setminus F'} w(e) + \sum_{e \in F' \setminus F} w'(e). \end{aligned}$$

We note that $d_w((F, w), (F', w)) = \sum_{e \in F \triangle F'} w(e)$ holds.

To understand the difference between the unweighted and weighted mappings, consider the shortest path problem. In this problem, a graph and a weight vector model a road network and the time required to pass through roads, respectively, and the output path represents the roads used in a trip. The unweighted distance d_u measures the number of roads changed between two trips, while the weighted distance d_w measures the time spent on different roads between the two trips.

The weighted mapping is more natural than the unweighted one for Lipschitzness. To see this, let us consider a shortest path algorithm $\mathcal{A}$. It is natural to ask $\mathcal{A}$ to output the same path regardless of whether the distance is measured in kilometers or miles. This implies that $\mathcal{A}$ is *scale invariant*, that is, it outputs the same path when edge weights are multiplied by a constant. Let $G = (V, E)$ be a graph and $w \in \mathbb{R}^E_{\geq 0}$ be a weight vector measured in kilometers, and consider another weight vector $w' \in \mathbb{R}^E_{\geq 0}$ obtained from w by setting $w'(e) = w(e) + \delta$, where $e \in E$ and $\delta > 0$ is measured in kilometers. Then, if we measure the distance between outputs using the weighted mapping, the relative change of the output is $d_w((\mathcal{A}(G, w), w), (\mathcal{A}(G, w'), w'))/\delta$. Let $c \approx 1.609$ be the ratio of a mile to a kilometer. Then, if we calculate edge weights in miles, the relative change of the output is

$$\begin{aligned} \frac{d_w((\mathcal{A}(G, w/c), w/c), (\mathcal{A}(G, w'/c), w'/c))}{\delta/c} &= \frac{d_w((\mathcal{A}(G, w), w), (\mathcal{A}(G, w'), w'))/c}{\delta/c} \\ &= \frac{d_w((\mathcal{A}(G, w), w), (\mathcal{A}(G, w'), w'))}{\delta}, \end{aligned}$$

and hence the two relative changes coincide. We do not have this property if we use the unweighted mapping. Hence, we first focus on the weighted mapping and then discuss the unweighted one later.

First Attempt: Lipschitz continuity of deterministic algorithms. Using the metric based on the weighted mapping imposed on the input and output spaces as mentioned previously, we can define the Lipschitz constant of a deterministic algorithm as follows:

Definition 1 (Lipschitz constant of a deterministic algorithm) *Let $\mathcal{A}$ be a deterministic algorithm that, given a graph $G = (V, E)$ and a weight vector $w \in \mathbb{R}^E_{\geq 0}$, outputs an edge set $\mathcal{A}(G, w) \subseteq E$. Then, the* Lipschitz constant *of the algorithm $\mathcal{A}$ on a graph $G = (V, E)$ is*

$$\sup_{\substack{w, w' \in \mathbb{R}^E_{\geq 0}, \\ w \neq w'}} \frac{d_w((\mathcal{A}(G, w), w), (\mathcal{A}(G, w'), w'))}{\|w - w'\|_1}.$$

Note that we only take the supremum over weight vectors and not over underlying graphs. To explain why we adopt this definition, let us consider the shortest path problem again. The weight vector can frequently change owing to traffic jams or inclement weather, whereas the underlying graph may change because of construction or disasters, which occur less frequently. Hence, it would be more useful to consider the former type of changes than the latter.

Another reason for not taking the supremum over pairs of graphs is that the change in the underlying graph often forces any (reasonable) algorithm to change its output drastically, and hence it is impossible to bound the Lipschitz constant if we allow changes in the underlying graph. For example, consider an instance of the shortest path problem such that there are two disjoint paths—one short and the other long—between source and target vertices. Any algorithm with a reasonable approximation guarantee must output the shorter path. However, if an edge in the shorter path is removed, the algorithm must change its output to the longer path.

Unfortunately, even though we do not take the supremum over underlying graphs in Definition 1, any (reasonable) deterministic algorithm for the shortest path problem is not Lipschitz continuous:

Definition 2 *Any deterministic algorithm for the shortest path problem with a finite approximation ratio is not Lipschitz continuous, that is, its Lipschitz constant is unbounded.*

To see the reason, consider a graph having two disjoint paths between the source and target vertices and the transition from a weight vector for which the first path is shorter to one for which the second path is shorter. Because the algorithm is deterministic, there is some point in the transition where the output path discontinuously changes from the first path to the second one, which implies that the algorithm is not Lipschitz.

Second Attempt: Lipschitz continuity of randomized algorithms. To remedy the aforementioned issue, we consider the Lipschitz continuity of randomized algorithms. First, we extend d_w, which is a metric over outputs, to a metric over output distributions. For two probability distributions $\mathcal{F}, \mathcal{F}'$ over subsets of E, the *earth mover's distance* between $\mathcal{F}$ and $\mathcal{F}'$ is defined as

$$\mathrm{EM}_{\mathrm{w}}\left((\mathcal{F}, w), (\mathcal{F}', w')\right) := \min_{\mathcal{D}} \mathop{\mathbb{E}}_{(F, F') \sim \mathcal{D}} d_w\left((F, w), (F', w')\right),$$

where the minimum is taken over *couplings* of $\mathcal{F}$ and $\mathcal{F}'$, that is, distributions over pairs of sets such that its marginal distributions on the first and second coordinates are equal to $\mathcal{F}$ and $\mathcal{F}'$, respectively. We note that EM_{w} coincides with d_w if the distributions $\mathcal{F}$ and $\mathcal{F}'$ are supported by single edge sets.

For a randomized algorithm $\mathcal{A}$, a graph $G = (V, E)$, and a weight vector $w \in \mathbb{R}^E_{\geq 0}$, let $\mathcal{A}(G, w)$ denote the (random) output of $\mathcal{A}$ on G and w. Abusing the notation, we often identify it with its distribution. Then, we define the Lipschitz constant of a randomized algorithm as follows:

Definition 3 (Lipschitz constant of a randomized algorithm) *Let $\mathcal{A}$ be a randomized algorithm that, given a graph $G = (V, E)$ and a weight vector $w \in \mathbb{R}^E_{\geq 0}$, outputs a (random) edge set $\mathcal{A}(G, w) \subseteq E$.*

Table 1: Results for Lipschitz continuity. n represents the number of vertices in the input graph, and $\epsilon, \alpha \in (0, 1)$ are arbitrary constants.

Problem	Approximation Ratio	Lipschitz Constant
Minimum Spanning Tree	$1+\epsilon$	$O(\epsilon^{-1})$
	$1+\epsilon$	$\Omega(\epsilon^{-1})$
Shortest Path	$1+\epsilon$	$O(\epsilon^{-1}\log^3 n)$
	$1+\epsilon$	$\Omega(\epsilon^{-1})$
Maximum Weight Matching	$1/8-\epsilon$	$O(\epsilon^{-1})$
	α	$\Omega(\alpha)$

Then, the Lipschitz constant *of the algorithm $\mathcal{A}$ on a graph $G=(V,E)$ is*

$$\sup_{\substack{w,w'\in\mathbb{R}^E_{\geq 0},\\ w\neq w'}} \frac{\mathrm{EM_w}\left((\mathcal{A}(G,w),w),(\mathcal{A}(G,w'),w')\right)}{\|w-w'\|_1}.$$

We say that $\mathcal{A}$ is Lipschitz continuous *if its Lipschitz constant is bounded for any graph $G=(V,E)$ and is* L-Lipschitz *if its Lipschitz constant is at most L.*

If the algorithm is deterministic, then this definition coincides with Definition 1.

Consider again the graph having two disjoint paths between the source and target vertices and the transition from a weight vector for which the first path is shorter to one for which the second path is shorter. Then, the output of a randomized algorithm can also make a transition from a distribution with most of its mass on the first path to one with most of its mass on the second path, and hence we can alleviate the issue of deterministic algorithms. However, designing Lipschitz continuous algorithms is a nontrivial task because we need to bound the ratio in Definition 3 for any pair of weight vectors, which can be very close.

1.3 Lipschitz Continuous Algorithms for Graph Problems

In this section, we discuss the Lipschitz continuity of randomized algorithms for several graph problems. Our results are summarized in Table 1.

Minimum spanning tree. In the *(weighted) minimum spanning tree problem*, we are given a (connected) undirected graph $G=(V,E)$, and a weight vector $w \in \mathbb{R}^E_{\geq 0}$, and the goal is to output a spanning tree $T \subseteq E$ that minimizes the total weight $\sum_{e\in T} w(e)$. We show that for any $\epsilon > 0$, there exists a polynomial-time $(1+\epsilon)$-approximation algorithm for the minimum spanning tree problem with Lipschitz constant $O(\epsilon^{-1})$. To understand this upper bound, suppose that w is $\{0,1\}$-valued and that we change the value of $w(f)$ from 1 to 0 for some edge $f \in E$. This is essentially equivalent to contracting the edge f, and the upper bound indicates that we only need to change $O(\epsilon^{-1})$ edges in the spanning tree, which is far smaller than the spanning tree size, i.e., $\Theta(n)$. We complement the upper bound by showing that any (randomized) $(1+\epsilon)$-approximation algorithm for the minimum spanning tree problem must have Lipschitz constant $\Omega(\epsilon^{-1})$.

Shortest path. In the *(weighted) shortest path problem*, we are given an undirected graph $G=(V,E)$, two vertices $s,t \in V$, and a weight vector $w \in \mathbb{R}^E_{\geq 0}$, and the goal is to output the shortest path between s and t, where the length of a path $P \subseteq E$ is $\sum_{e\in P} w(e)$. We show that for any $\epsilon \in (0,1)$, there exists a polynomial-time $(1+\epsilon)$-approximation algorithm for the shortest path problem with Lipschitz constant

$O\left(\epsilon^{-1}\log^3 n\right)$, where n represents the number of vertices in the input graph. Our algorithm may output a walk, i.e., the same edge may be used a multiple times in the output. We regard a walk P as a multiset of edges, and we map it to a vector $\sum_{e\in P} w(e)\mathbf{1}_e$, where an edge $e \in E$ appears in the sum the same number of times that it appears in the walk P. Then, the distance $d_w(\cdot,\cdot)$ for walks and the Lipschitz constant of an algorithm that outputs a walk can be naturally defined. To understand the upper bound, suppose that w is $\{0,1\}$-valued and that we change the value of $w(f)$ from 1 to 0 for some edge $f \in E$. This is essentially equivalent to contracting the edge f, and the upper bound indicates that we only need to change $O(\epsilon^{-1}\log^3 n)$ edges in the output path, which is nontrivially small when the shortest path from s to t is $\omega(\log^3 n)$. We also show that any (randomized) $(1+\epsilon)$-approximation algorithm for the shortest path problem must have Lipschitz constant $\Omega(\epsilon^{-1})$, which implies that our upper bound is tight up to a polylogarithmic factor in n.

Maximum weight matching. In the *maximum weight matching problem*, given a graph $G = (V, E)$ and a weight vector $w \in \mathbb{R}^E_{\geq 0}$, we want to find a matching $M \subseteq E$ with the maximum weight, i.e., $\sum_{e\in M} w(e)$. We show that for any $\epsilon > 0$, there exists a polynomial-time $(1/8-\epsilon)$-approximation algorithm with Lipschitz constant $O(\epsilon^{-1})$. To understand this upper bound, suppose again that w is $\{0,1\}$-valued and that we change the value of $w(f)$ from 1 to 0 for some $f \in E$. This is essentially equivalent to deleting the edge f, and the upper bound indicates that we only need to change $O(\epsilon^{-1})$ edges in the matching, which is nontrivially small when the matching size is $\omega(1)$. We also show that any (randomized) α-approximation algorithm for the maximum weight matching problem must have Lipschitz constant $\Omega(\alpha)$.

We note that the proof of Theorem 2 can be easily extended to the minimum spanning tree problem and the maximum weight matching problem, and hence randomness is necessary to obtain Lipschitz continuous algorithms for them.

1.4 Pointwise Lipschitz Continuity for Unweighted Mapping

In this section, we discuss Lipschitz continuity in the case where the distances between outputs are measured using the unweighted mapping. First, we define the earth mover's distance between output distributions $\mathcal{F}$ and $\mathcal{F}'$ with respect to the unweighted mapping as follows:

$$\mathrm{EM}_{\mathrm{u}}\left(\mathcal{F},\mathcal{F}'\right) := \min_{\mathcal{D}} \mathop{\mathbb{E}}_{(F,F')\sim\mathcal{D}} d_u(F,F'),$$

where the minimum is taken over couplings of $\mathcal{F}$ and $\mathcal{F}'$.

We cannot hope that a scale-invariant algorithm has a bounded Lipschitz constant with respect to the unweighted mapping. To see this, let $w, w' \in \mathbb{R}^E_{\geq 0}$ be arbitrary weight vectors. Then for any constant $c > 0$, we have

$$\frac{d_u(\mathcal{A}(G,w/c),\mathcal{A}(G,w'/c))}{\|w/c - w'/c\|_1} = \frac{c\cdot d_u(\mathcal{A}(G,w),\mathcal{A}(G,w))}{\|w-w'\|_1},$$

which implies that the Lipschitz constant is unbounded. Hence, we consider the following variant that look at the relative change in a local neighborhood:

Definition 4 (Pointwise Lipschitz constant of a randomized algorithm with respect to the unweighted mapping) *Let $\mathcal{A}$ be a randomized algorithm that, given a graph $G = (V, E)$ and a weight vector $w \in \mathbb{R}^E_{\geq 0}$, outputs a (random) edge set $\mathcal{A}(G, w) \subseteq E$. Then, the* pointwise Lipschitz constant *of the algorithm $\mathcal{A}$ on a graph $G = (V, E)$ at a weight vector $w \in \mathbb{R}^E_{\geq 0}$ with respect to the unweighted mapping is*

$$\limsup_{w'\in\mathbb{R}^E_{\geq 0}, w'\to w} \frac{\mathrm{EM}_{\mathrm{u}}\left(\mathcal{A}(G,w),\mathcal{A}(G,w')\right)}{\|w-w'\|_1}.$$

Table 2: Results for pointwise Lipschitz continuity with respect to the unweighted mapping. For the minimum spanning tree problem, n represents the number of vertices in the input graph, and for the maximum weight bipartite matching problem, n and m represent the number of vertices in the left and right parts of the input bipartite graph, respectively. $\epsilon \in (0, 1)$ is an arbitrary constant, and opt represents the optimal value.

Problem	Approximation Ratio	Lipschitz Constant
Minimum Spanning Tree	$1 + \epsilon$	$O(\epsilon^{-1} n/\mathrm{opt})$
Maximum Weight Bipartite Matching	$1/2 - \epsilon$	$O(\epsilon^{-1} n^{3/2} \log m/\mathrm{opt})$

In contrast to Lipschitz constant, the pointwise Lipschitz constant can depend on the weight vector w, and hence a scale-invariant algorithm can have a bounded pointwise Lipschitz constant.

We consider the pointwise Lipschitz continuity of algorithms with respect to the unweighted mapping for the minimum spanning tree problem and the maximum weight bipartite matching problem. Our results are summarized in Table 2. Below, we discuss them in detail.

For any $\epsilon > 0$, we show that there exists a polynomial-time $(1 + \epsilon)$-approximation algorithm for the minimum spanning tree problem with pointwise Lipschitz constant $O(\epsilon^{-1} n/\mathrm{opt})$, where n is the number of vertices in the input graph and opt is the minimum weight of a spanning tree. As discussed previously, the dependency on opt (or some other function depending on edge weights) is unavoidable. Suppose $w \in \mathbb{R}^E_{\geq 0}$ is $\{0, 1\}$-valued. Then, the bound shows that the change in the output tree is smaller than the tree size, $n - 1$, when $\mathrm{opt} = \omega_n(\epsilon^{-1})$.

In the *maximum weight bipartite matching problem*, given a complete bipartite graph $G = (U \cup V, E = U \times V)$ and a weight vector $w \in \mathbb{R}^E_{\geq 0}$, the goal is to output a matching $M \subseteq E$ that maximizes its weight, i.e., $\sum_{e \in M} w(e)$. For this problem, we show that there exists a polynomial-time $(1/2 - \epsilon)$-approximation algorithm with pointwise Lipschitz constant $O(\epsilon^{-1} n^{3/2} \log m/\mathrm{opt})$, where n and m are the numbers of vertices in the left and right parts of the input bipartite graph, respectively, and opt is the maximum weight of a matching. Suppose $n < m$ and the weight vector $w \in \mathbb{R}^E_{\geq 0}$ is $\{0, 1\}$-valued. Then, the bound shows that the change in the output matching is smaller than the maximum matching size, n, when $\mathrm{opt} = \omega(\epsilon^{-1} \sqrt{n} \log m)$.

We note that, in general, a Lipschitz continuous algorithm does not imply an algorithm with a bounded pointwise Lipschitz constant, and vice versa.

1.5 Related Work

Worst-case and average sensitivity. Lipschitz continuity is closely related to the *sensitivity* of algorithms introduced in [12, 16]. The *worst-case and average sensitivities* of a randomized algorithm $\mathcal{A}$ on an (unweighted) graph $G = (V, E)$ are defined as

$$\max_{e \in E} \mathrm{EM}_u(\mathcal{A}(G), \mathcal{A}(G \setminus e)) \quad \text{and} \quad \frac{1}{|E|} \sum_{e \in E} \mathrm{EM}_u(\mathcal{A}(G), \mathcal{A}(G \setminus e)), \tag{1}$$

respectively, where $G \setminus e$ is the graph obtained from G by deleting the edge $e \in E$. Clearly, the average sensitivity is bounded from above by the worst-case sensitivity. The sensitivity of algorithms has been investigated for various graph problems including the minimum cut problem [16], the maximum matching problem [16, 22], and spectral clustering [13]. It is known that there is no algorithm with $o(n)$ worst-case/average sensitivity for the shortest path problem [16]. As the definition of sensitivity (1) can be easily generalized, other non-graph problems such as dynamic programming problems [8, 9] and Euclidean k-means [20] have also been studied from the viewpoint of sensitivity.

To see the connection to Lipschitz continuity, suppose that in the supremum of Definition 3, we fix w to be the all-one vector $\mathbf{1}_E$ and we restrict the domain of w' to $\{0, 1\}$-valued vectors. Then by the

triangle inequality, the (modified) Lipschitz constant on a graph $G = (V, E)$ can be bounded from above as

$$\max_{w' \in \{0,1\}^E} \frac{\mathrm{EM}_{\mathrm{w}}((\mathcal{A}(G, \mathbf{1}_E), \mathbf{1}_E), (\mathcal{A}(G, w'), w'))}{\|\mathbf{1}_E - w'\|} = \max_{F \subseteq E} \frac{\mathrm{EM}_{\mathrm{w}}((\mathcal{A}(G, \mathbf{1}_E), \mathbf{1}_E), (\mathcal{A}(G, \mathbf{1}_{E \setminus F}), \mathbf{1}_{E \setminus F}))}{|F|}$$
$$\leq \max_{e \in E} \mathrm{EM}_{\mathrm{w}}((\mathcal{A}(G, \mathbf{1}_E), \mathbf{1}_E), (\mathcal{A}(G, \mathbf{1}_{E \setminus \{e\}}), \mathbf{1}_{E \setminus \{e\}})), \tag{2}$$

where $\mathbf{1}_{E \setminus \{e\}} \in \{0, 1\}^E$ is the characteristic vector of $E \setminus \{e\}$.

For the shortest path problem, the weighted graph (G, w) for a $\{0, 1\}$-valued weight vector w is equivalent to the graph obtained from G by contracting edges $e \in E$ with $w(e) = 0$. In particular, the weighted graph $(G, \mathbf{1}_{E \setminus \{e\}})$ is equivalent to G/e, which is the graph obtained from G by contracting the edge e. Hence, (2) can be seen as a variant of the worst-case sensitivity, where the operation applied to the graph is edge contraction instead of edge deletion, that is,

$$\max_{e \in E} \mathrm{EM}_u(\mathcal{A}(G), \mathcal{A}(G/e)). \tag{3}$$

Indeed, our Lipschitz continuous algorithm for the shortest path problem is based on an algorithm with a bounded sensitivity with respect to edge contraction.

A notable difference between the sensitivity and the Lipschitz constant is that the former is trivially bounded by the maximum solution size whereas it is not clear a priori whether there is an algorithm for which the latter is bounded.

Lipschitz continuity of neural networks. As mentioned previously, adversarial attacks is a considerable threat to machine learning-based systems. To mitigate the effects of adversarial attacks, neural networks with small Lipschitz constants have been proposed and their properties have been investigated [3, 5, 15, 17, 18]. It is also reported that bounding Lipschitz constants of neural networks stabilizes the training process and often produces models with better output quality [7, 11, 21].

We note that bounding the Lipschitz constant of a neural network is often easy because it is bounded by the product of the Lipschitz constants of the activation functions and linear transformations used in the neural network, which are easy to calculate. However, to design Lipschitz continuous algorithms for graph problems, we need to bound approximation ratio and Lipschitz constant simultaneously, and we often need nontrivial techniques as we will see in this paper.

1.6 Technical Overview

Minimum spanning tree. It is known that Kruskal's algorithm has (worst-case) sensitivity $O(1)$ against edge deletions [16]. However, it is not Lipschitz continuous because it is deterministic (see Theorem 2), and hence some modification is required.

Our Lipschitz continuous algorithm for the minimum spanning tree problem works as follows: Given a graph $G = (V, E)$ and a weight vector $w \in \mathbb{R}^E_{\geq 0}$, we sample $\widehat{\boldsymbol{w}}(e)$ uniformly from $[\boldsymbol{w}(e), (1+\epsilon)\boldsymbol{w}(e)]$ for each edge $e \in E$, and then apply Kruskal's algorithm to the new weight vector $\widehat{\boldsymbol{w}}$. This algorithm clearly achieves $(1 + \epsilon)$-approximation.

We can show that, to bound the Lipschitz constant, it suffices to consider a pair of weight vectors (w, w') such that w' is obtained from w by setting $w'(e) = w(e) + \delta$ for some $e \in E$ and $\delta > 0$. Then, the total variation distance between $\widehat{\boldsymbol{w}}$ and $\widehat{\boldsymbol{w}}'$ is $O(\epsilon^{-1}\delta/w(e))$, where $\widehat{\boldsymbol{w}}'$ is constructed from w' in the same way as $\widehat{\boldsymbol{w}}$ is constructed from w. Then, we can define a coupling, i.e., a joint distribution, between $\widehat{\boldsymbol{w}}$ and $\widehat{\boldsymbol{w}}'$ such that $\widehat{\boldsymbol{w}} \neq \widehat{\boldsymbol{w}}'$ with probability $O(\epsilon^{-1}\delta/w(e))$. Also, we can show that when $\boldsymbol{w} \neq \boldsymbol{w}'$ occurs in the coupling, the distance between the output spanning trees is $O(w(e))$. This implies that the earth mover's distance is $O(\epsilon^{-1}\delta)$ and hence the Lipschitz constant is $O(\epsilon^{-1})$. Our algorithm and analysis for pointwise Lipschitzness with respect to the unweighted mapping is similar though we need some care because we use the optimal value to determine the range from which we sample $\widehat{\boldsymbol{w}}(e)$ and it varies depending on the weight vector.

Shortest path. To obtain Lipschitz continuous algorithm for the shortest path problem, we first design an algorithm for the (unweighted) shortest path problem with a low sensitivity with respect to edge contraction (see (3)). We will use this algorithm as a subroutine in our Lipschitz continuous algorithm. The subroutine takes an unweighted graph $\widehat{G}$ and two vertices $s, t \in V(G)$ as the input and returns an approximate s-t shortest path in $\widehat{G}$. This subroutine is recursive: It samples a vertex $\boldsymbol{v}$ called a *pivot*, recursively computes s-$\boldsymbol{v}$ and $\boldsymbol{v}$-t walks that are nearly optimal, and then returns the walk obtained by concatenating the two walks. We choose the pivot $\boldsymbol{v}$ so that it is roughly in the middle of a nearly optimal s-t path. By doing so, we can bound the depth of the recursion by $O(\log n)$ and guarantee that the output walk is nearly optimal. Now, we turn to analyzing the sensitivity of the subroutine with respect to edge contraction. Let $e \in E(\widehat{G})$ and we want to bound the earth mover's distance between the output distributions for $\widehat{G}$ and $\widehat{G}/e$. Informally, we say that a recursion call for computing a u-v shortest path is *active* if there is a nearly optimal u-v path passing through e and is *inactive* otherwise. Then, we can show the following:

- recursion calls invoked in an inactive recursion call are also inactive, and
- with high probability, at most one of the two recursion calls invoked in an active recursion call is active.

These properties imply that the expected number of active recursion calls in each recursion depth is $O(1)$. Also, we can prove that in an inactive call, the pivot is sampled from exactly the same distributions for $\widehat{G}$ and $\widehat{G}/e$, and thus it does not contribute to the sensitivity at all. Additionally, we can show that each active call contributes to the sensitivity by $O(\log^2 n)$. Regarding that the number of active calls is $O(\log n)$, the sensitivity of the algorithm can be bounded by $O(\log^3 n)$.

To obtain a Lipschitz continuous algorithm for the shortest path problem, we construct an unweighted graph $\widehat{\boldsymbol{G}}(w)$ from the input weighted graph (G, w), and apply the subroutine to it to obtain a walk in $\widehat{\boldsymbol{G}}(w)$, and then output the corresponding walk in G. Here, the graph $\widehat{\boldsymbol{G}}(w)$ is obtained by replacing each edge of the input graph G with a path of suitable length so that each path in G naturally corresponds to a path in $\widehat{\boldsymbol{G}}(w)$. The length of each path is sampled from a certain distribution so that the total variation distance between $\widehat{\boldsymbol{G}}(w)$ and $\widehat{\boldsymbol{G}}(w')$ is proportional to $\|w - w'\|_1$. Then, we can bound the Lipschitz continuity of the algorithm by using the sensitivity of the subroutine.

Maximum weight matching. Our algorithm is based on the algorithm for the maximum weight matching proposed by Yoshida and Zhou [22]. For a parameter $\alpha > 2$, their algorithm first classifies edges e according to the value $\lfloor \log_\alpha w(e) \rfloor$. Then, it runs a randomized greedy to compute a matching for each edge class, and then returns a matching obtained by combining them.

Although their algorithm has a bounded *weighted sensitivity*, a discrete analogue of Lipschitz constant, its Lipschitz constant is not bounded. This is because an arbitrarily small change in the weight of an edge may cause the edge to be classified into a different class. To resolve this issue, we sample a parameter $\boldsymbol{b} \in [1, \alpha]$ and classify edges e according to the value $\left\lfloor \log_\alpha \frac{w(e)}{\boldsymbol{b}} \right\rfloor$. Then, the total variation distance between the classifications obtained from weight vectors w and w' is proportional to $\|w - w'\|_1$, and we can bound the Lipschitz constant.

Maximum weight bipartite matching. In contrast to the previous Lipschitz continuous algorithm for the maximum weight matching problem, our pointwise Lipschitz continuous algorithm for the maximum weight bipartite matching problem is based on linear programming (LP). The standard LP relaxation for the maximum weight bipartite matching problem is not stable against perturbations to the edge weight. Hence, we consider LP with *entropy regularization* [4]. Although entropy regularization was originally introduced to speed up the computation of the earth mover's distance, we use it here to stabilize the computation of LP. For a weight vector $w \in \mathbb{R}^E_{\geq 0}$, let $\mathrm{LP}_{\mathrm{ent}}(w)$ denote the LP for the weighted graph (G, w) with entropy regularization. Then, we can show that (i) the solution to $\mathrm{LP}_{\mathrm{ent}}$ is nearly optimal to the original LP, and (ii) for any weight vector w', the ℓ_1 distance between the solutions of $\mathrm{LP}_{\mathrm{ent}}(w)$

and $\mathrm{LP}_{\mathrm{ent}}(w')$ is proportional to $\|w - w'\|_1$. Then, we carefully round the obtained fractional solution to an integral one in such a way that for any two fractional solutions x and x', the earth mover's distance between the (random) integer solutions obtained from x and x' is proportional to $\|x - x'\|_1$ with respect to the unweighted mapping.

References

[1] Naveed Akhtar and Ajmal Mian. Threat of adversarial attacks on deep learning in computer vision: A survey. *IEEE Access*, 6:14410–14430, 2018.

[2] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57. Ieee, 2017.

[3] Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. Parseval networks: Improving robustness to adversarial examples. In *International Conference on Machine Learning*, pages 854–863. PMLR, 2017.

[4] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

[5] Mahyar Fazlyab, Alexander Robey, Hamed Hassani, Manfred Morari, and George Pappas. Efficient and accurate estimation of lipschitz constants for deep neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

[6] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015.

[7] Henry Gouk, Eibe Frank, Bernhard Pfahringer, and Michael J Cree. Regularisation of neural networks by enforcing lipschitz continuity. *Machine Learning*, 110(2):393–416, 2021.

[8] Soh Kumabe and Yuichi Yoshida. Average sensitivity of dynamic programming. In *Proceedings of the 33th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1925–1961, 2022.

[9] Soh Kumabe and Yuichi Yoshida. Average sensitivity of the knapsack problem. In *30th Annual European Symposium on Algorithms (ESA)*, volume 244, pages 75:1–75:14, 2022.

[10] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2018.

[11] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018.

[12] Shogo Murai and Yuichi Yoshida. Sensitivity analysis of centralities on unweighted networks. In *Proceedings of the 2019 World Wide Web Conference (WWW)*, pages 1332–1342, 2019.

[13] Pan Peng and Yuichi Yoshida. Average sensitivity of spectral clustering. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, pages 1132–1140, 2020.

[14] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2013.

[15] Yusuke Tsuzuku, Issei Sato, and Masashi Sugiyama. Lipschitz-margin training: Scalable certification of perturbation invariance for deep neural networks. *Advances in neural information processing systems*, 31, 2018.

[16] Nithin Varma and Yuichi Yoshida. Average sensitivity of graph algorithms. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 684–703. 2021.

[17] Aladin Virmaux and Kevin Scaman. Lipschitz regularity of deep neural networks: analysis and efficient estimation. *Advances in Neural Information Processing Systems*, 31, 2018.

[18] Tsui-Wei Weng, Huan Zhang, Pin-Yu Chen, Jinfeng Yi, Dong Su, Yupeng Gao, Cho-Jui Hsieh, and Luca Daniel. Evaluating the robustness of neural networks: An extreme value theory approach. In *International Conference on Learning Representations*, 2018.

[19] Han Xu, Yao Ma, Hao-Chen Liu, Debayan Deb, Hui Liu, Ji-Liang Tang, and Anil K Jain. Adversarial attacks and defenses in images, graphs and text: A review. *International Journal of Automation and Computing*, 17(2):151–178, 2020.

[20] Yuichi Yoshida and Shinji Ito. Average sensitivity of Euclidean k-clustering. In *NeurIPS*, 2022. to appear.

[21] Yuichi Yoshida and Takeru Miyato. Spectral norm regularization for improving the generalizability of deep learning. *arXiv preprint arXiv:1705.10941*, 2017.

[22] Yuichi Yoshida and Samson Zhou. Sensitivity analysis of the maximum matching problem. In *Innovations in Theoretical Computer Science (ITCS)*, pages 58:1–58:20, 2021.

[23] Wei Emma Zhang, Quan Z Sheng, Ahoud Alhazmi, and Chenliang Li. Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM Transactions on Intelligent Systems and Technology*, 11(3):1–41, 2020.

Simultaneous Assignments

Péter Madarasi

Department of Operations Research, ELTE
Eötvös Loránd University, and the ELKH-ELTE
Egerváry Research Group on Combinatorial
Optimization, Eötvös Loránd Research Network
(ELKH), Pázmány Péter sétány 1/C, 1117
Budapest, Hungary.
madarasip@staff.elte.hu

Abstract: This paper introduces the *simultaneous assignment problem*. Let us given a graph with a weight and a capacity function on its edges, and a set of its subgraphs along with a degree upper bound function for each of them. In addition, we are given a laminar system on the nodes with an upper bound on the degree-sum of the nodes in each member of the system. Our goal is to assign each edge a non-negative integer below its capacity such that the total weight is maximized, the degrees in each subgraph are below the bound associated with the subgraph, and the degree-sum bound is respected in each member of the laminar system.

We identify special cases when the problem can be shown to be solvable in polynomial time. One of these cases is a common generalization of the *hierarchical b-matching problem* and the *laminar matchoid problem*. This implies that both problems can be solved efficiently in the weighted, capacitated case even if both lower and upper bounds are present — generalizing the previous polynomial-time algorithms. The problem is also solvable for trees provided that the laminar system is empty and a natural assumption holds for the subgraphs.

The general problem, however, is shown to be APX-hard in the unweighted case. Furthermore, we prove that the approximation guarantee of any polynomial-time algorithm must increase asymptotically linearly in the number of the given subgraphs unless P=NP.

We give a generic framework for deriving approximation algorithms, which can be applied to a wide range of problems. As an application to our problem, a constant-approximation algorithm is derived when the number of the given subgraphs is a constant. The approximation guarantee is the same as the integrality gap of a strengthened LP-relaxation when the number of the given subgraphs is small. Furthermore, improved approximation algorithms are given for special cases, for example, when the degree bounds are uniform or the graph is sparse.

Keywords: Restricted b-matching, Matchoid, Approximation algorithms, Integrality gap

1 Introduction

In the *simultaneous assignment problem*, we are given a graph with a weight and capacity function on its edges and a set of its subgraphs along with a degree upper bound function for each of them. We are

[1]Supported by the ÚNKP-22-4 New National Excellence Program of the Ministry for Culture and Innovation from the source of the National Research, Development and Innovation Fund. The work was supported by the Lendület Programme of the Hungarian Academy of Sciences – grant number LP2021-1/2021, and the Ministry of Innovation and Technology of Hungary from the National Research, Development and Innovation Fund, financed under the ELTE TKP 2021-NKTA-62 funding scheme.

also given a laminar system on the node set with an upper bound on the degree-sum of the nodes in each member of the system. Our goal is to assign each edge a non-negative integer below its capacity such that the total weight is maximized, the degrees in each subgraph are below the degree upper bound associated with the subgraph, and the degree-sum bound is respected in each member of the laminar system.

More precisely, given are 1) a loop-free graph $G = (V, E)$, 2) a capacity function $c : E \to \mathbb{Z}_+$, 3) a set $\mathcal{H}$ of the subgraphs of G along with a function $b_H : V_H \to \mathbb{Z}_+$ for each subgraph $H = (V_H, E_H) \in \mathcal{H}$ and 4) a laminar system $\mathcal{L}$ on the nodes of G with a function $g : \mathcal{L} \to \mathbb{Z}_+$. An integer vector $x \in \mathbb{Z}_+^E$ is a *simultaneous assignment* if 1) $x \leq c$, 2) x is a c-capacitated b_H-matching in H for all $H \in \mathcal{H}$ and 3) $\sum_{v \in L} \sum_{e \in \Delta_G(v)} x_e \leq g(L)$ holds for all $L \in \mathcal{L}$, where $\Delta_G(v)$ denotes the set of edges incident to node v. In the weighted version of the problem, $\sum_{e \in E} w_e x_e$ is to be maximized for a given weight function $w : E \to \mathbb{R}_+$. The case $w \equiv 1$ will be referred to as the *unweighted* problem. We will see that this cumbersome problem includes a number of natural, interesting special cases — some of which are solvable in polynomial time.

Note that constraint 3) poses an upper bound on the x-degree sum in each $L \in \mathcal{L}$ — this means that the x-values of the edges induced by L count twice while those of the non-induced incident edges once. Constraints 3) will be referred to as the *degree-sum constraints*. The problem is called *uncapacitated* when $c \equiv \infty$. Without loss of generality, we assume that no subgraph in $\mathcal{H}$ contains isolated nodes.

The integer solutions of the following linear program are, by definition, the feasible simultaneous assignments.

$$\max \sum_{st \in E} w_{st} x_{st} \tag{LP1}$$

s.t.

$$x \in \mathbb{R}_+^E \tag{1a}$$

$$x_e \leq c_e \qquad \forall e \in E \tag{1b}$$

$$\sum_{e \in \Delta_H(v)} x_e \leq b_H(v) \qquad \forall H \in \mathcal{H}\ \forall v \in V_H \tag{1c}$$

$$\sum_{v \in L} \sum_{e \in \Delta_G(v)} x_e \leq g(L) \qquad \forall L \in \mathcal{L} \tag{1d}$$

Indeed, all feasible integer solutions to (LP1) are non-negative by (1a) and respect the capacity constraints by (1b). The degree constraints in each subgraph $H \in \mathcal{H}$ and the degree-sum constraint for $L \in \mathcal{L}$ hold by (1c) and (1d), respectively.

Motivation Imagine k consecutive events taking place in the same hall and a set of attendees who want to buy tickets. Given the event and the seat, we know how much a ticket costs. Each customer provides the list of seats that would suit him/her and also selects which of the k events (possibly more than one) they want to attend. Our goal is to assign customers to seats such that the total income is maximized and if somebody wanted to attend multiple events, then he/she must be either completely refused or seated to the same place for all the events. This scenario can be modeled as a simultaneous assignment problem as follows. Define a bipartite graph $G = (S, T; E)$, where S corresponds to the customers and T to the seats. For $s \in S$ and $t \in T$, add st to E if customer s likes seat t, and let $w(st)$ be the income if s is seated to t. Let S_i denote the set of customers who want to attend event i, and let $\mathcal{H} = \{H_1, \ldots, H_k\}$, where H_i is the subgraph of G induced by node set $S_i \cup T$. Let $c \equiv 1$, $\mathcal{L} = \emptyset$ and $b_{H_i} \equiv 1$ for $i \in \{1, \ldots, k\}$. By the construction, there is a one-to-one correspondence between feasible customer-seat assignments and feasible simultaneous assignments.

It is quite natural to require these constraints only for intervals of events, that is, if a customer skips some of the events, then he/she may be seated to a new place when he/she arrives back. Observe that a participant s leaving the hall at some point can be replaced with new dummy participants each of whom attends exactly one of the intervals of the events selected by s. That is, one can assume that each customer participates in an interval of events. This special case will be investigated in Section 2.2.

Previous work In the special case when $\mathcal{H} = \{G\}$ and $\mathcal{L} = \emptyset$, one gets back the usual *weighted capacitated b-matching problem*, where $b = b_G$ [1]. Another way to obtain this problem is when $\mathcal{H} = \emptyset$ and $\mathcal{L} = \{\{v\} : v \in V\}$, where $b(v) = g(\{v\})$ for all $v \in V$.

For the *ℓ-matchiod problem*, there exists an FPT algorithm parameterized by ℓ and the size of the solution [5]. This immediately implies that for the simultaneous assignment problem with $\mathcal{L} = \emptyset$ and $c \equiv 1$, there exists an FPT algorithm parameterized by the size of the solution and the size of $\mathcal{H}$.

One can show that uncapacitated simultaneous assignments form a $(2|\mathcal{H}|+1)$-*extendible system*, hence the greedy algorithm is a $(2|\mathcal{H}|+1)$-approximation algorithm [10]. This result is not hard to extend to the capacitated version, hence one gets that there exists a $(2|\mathcal{H}|+1)$-approximation algorithm for the simultaneous assignment problem.

In the *double matching problem*, we are given a bipartite graph $G = (S, T; E)$ and $S_1, S_2 \subseteq S$ such that $S_1 \cup S_2 = S$. It is NP-complete to decide whether there exists $M \subseteq E$ for which $|M| = |S|$ and both $M \cap E_1$ and $M \cap E_2$ are matchings, where E_i denotes the edges induced by T and S_i for $i \in \{1, 2\}$ [8]. The double matching problem is a special case of the simultaneous assignment problem, and this implies that it is NP-complete to decide whether a simultaneous assignment satisfying constraints (1c) with equality exists, even if $\mathcal{L} = \emptyset$.

As a direct application of the bounded-violation algorithms given for the *upper bounded degree g-polymatroid element problem* described in [2], one can find a vector $z \in \mathbb{Z}^E$ in polynomial time such that wz is at least the weight of the optimal simultaneous assignment, and it satisfies constraints (1a) and (1b), but it may violate constraints (1c) by an additive factor of at most $(2|\mathcal{H}| - 1)$, provided that $\mathcal{L} = \emptyset$.

Our results The special case when $\mathcal{H} = \emptyset$ corresponds to the so-called *weighted hierarchical b-matching problem*. This problem was introduced in [3], where a strongly polynomial-time algorithm was given for the unweighted case. Answering an open question from the same paper, our results in Section 2.1 imply that the weighted version of the problem can be solved in strongly polynomial time as well.

If $\mathcal{L} = \emptyset$ and $\mathcal{H}$ is such that the subgraphs in $\mathcal{H}$ restricted to the edges incident to v form a laminar system for each node $v \in V$, then we get back the *laminar matchoid problem* [6]. In [7], the laminar matchoid problem was solved in polynomial time when the so-called similarity condition holds, that is, the components of b and c are polynomial in the size of V. Our results in Section 2.1 also imply that the problem can be solved in strongly polynomial time even if the similarity condition does not hold.

In fact, Section 2.1 solves the simultaneous assignment problem in strongly polynomial time when $\mathcal{H}$ is such that the subgraphs in $\mathcal{H}$ restricted to the edges incident to v form a laminar system for each node $v \in V$. This can be seen as a common generalization of the (weighted, capacitated) hierarchical b-matching and the laminar matchoid problems, in which the laminar matchoid problem subject to the degree-sum (or hierarchical) constraints is to be solved. This approach settles the weighted, capacitated version of this common generalization even if both lower and upper bounds are given on the degrees in the subgraphs in $\mathcal{H}$ and on the degree-sums in the members of $\mathcal{L}$ — generalizing the hierarchical b-matching and the laminar matchoid problems with the presence of capacities and both lower and upper bounds.

We show in Section 2.2 that the simultaneous assignment problem can be solved for trees when $\mathcal{L} = \emptyset$ and the so-called local-interval property holds — the latter corresponds to the special case of the first motivation above when each customer is supposed to participate in an interval of the events.

Section 3 proves the NP-hardness of α-approximating the unweighted problem on bipartite graphs in two special cases for small enough constant α: 1) the size of $\mathcal{H}$ is two and all connected components of G are claws 2) each subgraph in $\mathcal{H}$ consists of two edges, the size of all members of $\mathcal{L}$ is at most two and all connected components of G are claws. Furthermore, we also show that the approximation guarantee of any polynomial-time approximation algorithm must grow asymptotically linearly in the size of $\mathcal{H}$.

Then, Section 4 introduces the concept of (m, ℓ)-covers and gives a general framework for deriving approximation algorithms, which can be applied to any problem in which one has an efficiently solvable special case with which every instance of the problem can be covered "equitably". The rest of Section 4 applies this approach to the simultaneous assignment problem. First, Section 4.2 gives an approximation algorithm when the size of $\mathcal{H}$ is small, and gives a bound on the integrality gap of a strengthened

LP-relaxation. In Section 4.1, we give an improved algorithm for the uniform case, that is, when the coordinates of all the degree bounds b_H are the same.

Notation Throughout this paper, $G = (V, E)$ is an undirected loop-free graph. Let $N_G(v)$ denote the set of the neighbors of v. For a subset X of the nodes, $\Delta_G(X)$ denotes the union of the edges incident to the nodes in X. We use $\deg_G(v)$ to denote the degree of node v in G. The maximum of the empty set is $-\infty$ by definition. Given a function $f : A \to B$, both $f(a)$ and f_a denote the value f assigns to $a \in A$, and let $f(X) = \sum_{a \in X} f(a)$ for $X \subseteq A$. Let χ_Z denote the characteristic vector of set Z, that is, $\chi_Z(y) = 1$ if $y \in Z$, and 0 otherwise. Occasionally, the braces around sets consisting of a single element are omitted, for example, $\Delta_G(\{v\}) = \Delta_G(v)$ for $v \in V$. The power set of a set X is denoted by 2^X. Let $\mathbb{N}$ and $\mathbb{Z}_+$ denote the sets of positive and non-negative integers, respectively.

2 Tractable Cases

2.1 Locally Laminar Subgraphs

A set system $\mathcal{F}$ is *laminar* if, for any two members $X, X' \in \mathcal{F}$, either $X \subseteq X'$, $X' \subseteq X$ or $X \cap X' = \emptyset$ holds. A set $\mathcal{H}$ of the subgraphs of G is *laminar* if the edge sets of the subgraphs in $\mathcal{H}$ form a laminar system. We say that $\mathcal{H}$ is *locally laminar* if, for each node $v \in V$, the subgraphs in $\mathcal{H}$ restricted to $\Delta_G(v)$ form a laminar system, that is, $\mathcal{F}_v = \{\Delta_H(v) : H \in \mathcal{H}\}$ is laminar for all $v \in V$. By definition, if $\mathcal{H}$ is laminar, then it is locally laminar.

Throughout this section, assume that $\mathcal{H}$ is locally laminar. In what follows, a polynomial-time algorithm is given to solve the weighted simultaneous assignment problem under this condition. The description of the polyhedron of feasible simultaneous assignments will be derived as well. The following definition and two theorems from [11, Page 594-598] will be useful.

Definition 1 *An integer matrix $M \in \mathbb{Z}^{m \times n}$ is* bidirected *if $\sum_{i=1}^{m} |M_{ij}| = 2$ for all $j \in \{1, \ldots, n\}$.*

For a matrix $M \in \mathbb{Z}^{m \times n}$ and vectors $a, b \in \mathbb{Z}^m$ and $c, d \in \mathbb{Z}^n$, we consider the integer solutions of

$$\{x \in \mathbb{Z}^n : d \leq x \leq c, a \leq Mx \leq b\}. \tag{2}$$

Theorem 2 *For a bidirected matrix $M \in \mathbb{Z}^{m \times n}$ and for arbitrary vectors $a, b \in \mathbb{Z}^m$ and $c, d \in \mathbb{Z}^n$, the convex hull of the integer solutions of (2) is described by the following system.*

(LP2)

$$x \in \mathbb{R}^n \tag{3a}$$

$$d \leq x \leq c \tag{3b}$$

$$a \leq Mx \leq b \tag{3c}$$

$$\frac{1}{2}((\chi_U - \chi_W)M + \chi_F - \chi_H)x \leq \left\lfloor \frac{1}{2}(b(U) - a(W) + c(F) - d(H)) \right\rfloor$$
$$\text{for all disjoint } U, W \subseteq \{1, \ldots, m\} \text{ and for all partition } F, H$$
$$\text{of } \delta(U \cup W) \text{ with } b(U) - a(W) + c(F) - d(H) \text{ odd,} \tag{3d}$$

where $\delta(U \cup W) = \{j \in \{1, \ldots, n\} : \sum_{i \in U \cup W} |M_{ij}| = 1\}$.

Theorem 3 *For a bidirected matrix $M \in \mathbb{Z}^{m \times n}$, and arbitrary vectors $a, b \in \mathbb{Z}^m$, $c, d \in \mathbb{Z}^n$ and $w \in \mathbb{Q}^n$, an integer vector x maximizing wx over (2) can be found in strongly polynomial time.*

It is not hard to show that Theorems 2 and 3 hold in the slightly more general case when $M \in \mathbb{Z}^{m \times n}$ is such that $\sum_{i=1}^{m} |M_{ij}| \leq 2$ for all $j \in [n]$:

Corollary 4 *Let $M \in \mathbb{Z}^{m\times n}$ be a matrix such that $\sum_{i=1}^{m} |M_{ij}| \leq 2$ for all $j \in \{1,\dots,n\}$, and let $a, b \in \mathbb{Z}^m$, $c, d \in \mathbb{Z}^n$. Then, the convex hull of the integer solutions of (2) is described by (LP2).*

Corollary 5 *Let $M \in \mathbb{Z}^{m\times n}$ be a matrix such that $\sum_{i=1}^{m} |M_{ij}| \leq 2$ for all $j \in \{1,\dots,n\}$, and let $d, c \in \mathbb{Z}^n, a, b \in \mathbb{Z}^m, w \in \mathbb{Q}^n$. Then, an integer vector x maximizing wx over (2) can be found in strongly polynomial time.*

In what follows, we show that the locally laminar simultaneous assignment problem can be formulated in such a way that it fits the framework given by (2), where M is such that $\sum_{i=1}^{m} |M_{ij}| \leq 2$ for all j — and hence one can apply Corollaries 4 and 5. First, consider the following notation. For a laminar system $\mathcal{F}$, let $\mathcal{C}(\mathcal{F})$ denote the inclusion-wise maximal sets in $\mathcal{F}$. The maximal sets in $\mathcal{F}$ inside a member X in $\mathcal{F}$ will be denoted by $\mathcal{C}(\mathcal{F}, X)$. For a degree-sum-constrained simultaneous assignment x, let $y^v_F = \sum_{e\in F} x_e$ for $v \in V$ and $F \in \mathcal{F}_v$. Furthermore, let $z_L = \sum_{v\in L} \sum_{e\in\Delta_G(v)} x_e$ for $L \in \mathcal{L}$. That is, y^v_F is the x-degree of node v restricted to F and z_L is the sum of the x-degrees of the nodes in L — which appear as the left-hand side of (1c) and (1d), respectively. By definition,

$$y^v_F = \sum_{e\in F\setminus\bigcup\mathcal{C}(\mathcal{F}_v,F)} x_e + \sum_{F'\in\mathcal{C}(\mathcal{F}_v,F)} y^v_{F'} \tag{4}$$

holds for all $v \in V$ and $F \in \mathcal{F}_v$. Without loss of generality, assume that $\{v\} \in \mathcal{L}$ for all $v \in V$ (if this is not the case for some $v \in V$, then one can add $\{v\}$ to $\mathcal{L}$ and set $g(\{v\}) = \infty$). Then,

$$z_{\{v\}} = \sum_{e\in\Delta_G(v)\setminus\bigcup\mathcal{C}(\mathcal{F}_v)} x_e + \sum_{F\in\mathcal{C}(\mathcal{F}_v)} y^v_F \tag{5}$$

holds for all $v \in V$ as well. Similarly to (4),

$$z_L = \sum_{v\in L\setminus\bigcup\mathcal{C}(\mathcal{L},L)} z_{\{v\}} + \sum_{L'\in\mathcal{C}(\mathcal{L},L)} z_{L'} \tag{6}$$

holds for all $L \in \mathcal{L}$. Considering x, y and z as variables, and combining (LP1) with equations (4), (5) and (6), one obtains the following linear program.

$$\max \sum_{st\in E} w_{st}x_{st} \tag{LP3}$$

s.t.

$$x \in \mathbb{R}^E_+ \tag{7a}$$
$$y^v \in \mathbb{R}^{\mathcal{F}_v}_+ \qquad \forall v \in V \tag{7b}$$
$$z \in \mathbb{R}^{\mathcal{L}}_+ \tag{7c}$$
$$x \leq c \tag{7d}$$
$$y^v_{\Delta_H(v)} \leq b_H(v) \qquad \forall H \in \mathcal{H}, v \in V_H \tag{7e}$$
$$z \leq g \tag{7f}$$
$$\sum_{e\in F\setminus\bigcup\mathcal{C}(\mathcal{F}_v,F)} x_e + \sum_{F'\in\mathcal{C}(\mathcal{F}_v,F)} y^v_{F'} - y^v_F = 0 \qquad \forall v \in V,\ \forall F \in \mathcal{F}_v \tag{7g}$$
$$\sum_{e\in\Delta_G(v)\setminus\bigcup\mathcal{C}(\mathcal{F}_v)} x_e + \sum_{F\in\mathcal{C}(\mathcal{F}_v)} y^v_F - z_{\{v\}} = 0 \qquad \forall v \in V \tag{7h}$$
$$\sum_{v\in L\setminus\bigcup\mathcal{C}(\mathcal{L})} z_{\{v\}} + \sum_{L'\in\mathcal{C}(\mathcal{L})} z_{L'} - z_L = 0 \qquad \forall L \in \mathcal{L}\setminus\{\{v\} : v \in V\} \tag{7i}$$

By the construction, the solutions to (LP3) restricted to x are exactly the feasible simultaneous assignments. Note that each variable appears at most twice in constraints (7g), (7h) and (7i) with coefficient 1 or -1, hence the matrix M given by these three sets of constraints is such that $\sum_{i=1}^{m} |M_{ij}| \leq 2$. Therefore, Corollary 5 immediately implies the following.

Theorem 6 *If $\mathcal{H}$ is locally laminar, then the simultaneous assignment problem can be solved in strongly polynomial time.*

As it has been already mentioned in Section 1, the hierarchical b-matching problem [3] is exactly the simultaneous assignment problem with $\mathcal{H} = \emptyset$. Since in this case $\mathcal{H}$ is locally laminar, Theorem 6 can be applied.

Corollary 7 *The weighted, capacitated hierarchical b-matching problem can be solved in strongly polynomial time.*

In the case when $\mathcal{L} = \emptyset$ and $\mathcal{H}$ is laminar, we get back the laminar matchoid problem, hence we obtain the following.

Corollary 8 *The weighted laminar matchoid problem can be solved in strongly polynomial time.*

In fact, Theorem 6 applies even if we pose both *lower* and upper bounds in constraints (7d), (7e) and (7f). This immediately implies the following:

Theorem 9 *The locally laminar case can be also solved when both* lower *and upper bounds are given in (1b), (1c) and (1d), that is, on the capacities of the edges, on the degrees in each subgraph $H \in \mathcal{H}$ and on the degree-sum in each $L \in \mathcal{L}$.*

Corollary 10 *The weighted, capacitated hierarchical b-matching problem can be solved in strongly polynomial time even when both* lower *and upper bounds are given on the capacities of the edges and on the degree sums in each member of the laminar family.*

Note that Theorem 2 gives a description of the convex hull of the integer points of (LP3). Substituting variables y and z, this in turn implies a description of the convex hull of the integer points of (LP1) when $\mathcal{H}$ is locally laminar. These constraints will be referred to as *projected blossom inequalities.*

Theorem 11 *If $\mathcal{H}$ is locally laminar, then (LP1) extended with the projected blossom inequalities describes the convex hull of simultaneous assignments.*

The projected blossom inequalities can be used to strengthen (LP1) if $\mathcal{H}$ is not locally laminar. Namely, one can add the projected blossom inequalities to (LP1) for all $F \subseteq E$ for which $\mathcal{H}$ becomes locally laminar when restricted to F:

$$\max \sum_{st \in E} w_{st} x_{st} \qquad \text{(LP1}^*\text{)}$$

s.t.

$(1a)$ $(1b)$ $(1c)$ $(1d)$

$$\mathcal{B}(F) \qquad \forall F \subseteq E : \mathcal{H}|_F \text{ is locally laminar,} \qquad \text{(8a)}$$

where $\mathcal{B}(F)$ denotes the set of projected blossom inequalities when the problem is restricted to $F \subseteq E$. By Theorem 11, the polyhedron defined by (LP1*) becomes integer when the problem is restricted to a locally laminar edge set. The integrality gap of this strengthened linear program will be investigated in Section 4.

2.2 When the Graph is a Tree

In Section 3, we will see that the simultaneous assignment problem is hard even if G consists of node-disjoint claws and the size of $\mathcal{H}$ is two. However, assuming that G is a tree and $\mathcal{L} = \emptyset$, the problem becomes solvable provided that a natural assumption on $\mathcal{H}$ holds. Motivated by the first application described in the introduction, consider the following definition.

Definition 12 *We say that $\mathcal{H} = \{H_1, \ldots, H_k\}$ has the* local-interval property *if, for each node $v \in V$, there exists a permutation $H_{i_1}, \ldots, H_{i_k}$ under which each edge in Δ_v is included in a (possibly empty) interval $H_{i_p}, \ldots, H_{i_q}$.*

Note that the local-interval property holds in the first motivation mentioned in the introduction when each customer selects an interval of the events. Under these assumptions, one can prove that the matrix of (LP1) is a network matrix [4, Page 151], hence the problem is solvable in polynomial time:

Theorem 13 *Let $G = (V, E)$ be a tree, let $\mathcal{L} = \emptyset$ and assume that $\mathcal{H}$ has the local-interval property. Then, the simultaneous assignment polyhedron is described by (LP1) and hence the problem can be solved in strongly polynomial time.*

Observe that $\mathcal{H}$ has the local-interval property if its size is two, therefore, one obtains the following corollary of Theorem 13.

Corollary 14 *Let G be a tree, let $\mathcal{L} = \emptyset$ and assume that the size of $\mathcal{H}$ is two. Then, the simultaneous assignment polyhedron is described by (LP1) and hence the problem can be solved in strongly polynomial time.*

3 Hardness Results

As we have already seen in the introduction, it is NP-complete to decide whether a simultaneous assignment satisfying constraints (1c) with equality exists. In this section, we give further hardness results, the proofs of which can be found in [9]

Theorem 15 *The unweighted simultaneous assignment problem is NP-hard to approximate within any factor smaller than $\frac{570}{569}$ even if $|\mathcal{H}| = 2$, the connected components of G are claws and the size of every member in $\mathcal{L}$ is two.*

One can also show that the problem is hard to approximate when the size of each subgraph in $\mathcal{H}$ is assumed to be two.

Corollary 16 *The unweighted simultaneous assignment problem is NP-hard to approximate within any factor smaller than $\frac{570}{569}$ even if all subgraphs in $\mathcal{H}$ consist of at most two edges, the size of all members of $\mathcal{L}$ is two and the connected components of G are claws.*

Similar results hold in the weighted case even if $\mathcal{L} = \emptyset$.

Theorem 17 *The weighted simultaneous assignment problem is NP-hard to approximate within any factor smaller than $\frac{760}{759}$, even if $|\mathcal{H}| = 2$, $\mathcal{L} = \emptyset$, each occurring weight is either 1 or 2 and G is bipartite with maximum degree at most four.*

It is quite natural to ask whether there exists an α-approximation algorithm for some constant α independent of the size of $\mathcal{H}$. The next theorem shows that no such algorithm is possible, in fact, the approximation factor must grow asymptotically (essentially) linearly with the size of $\mathcal{H}$ unless P=NP.

Theorem 18 *The simultaneous assignment problem is $\Omega(|\mathcal{H}|^{1-\epsilon})$-inapproximable for all $\epsilon > 0$ unless P=NP. The result holds even if $\mathcal{L} = \emptyset$.*

4 Approximation Algorithms

Throughout this section, let $k = |\mathcal{H}|$, $\mathcal{H} = \{H_1, \ldots, H_k\}$ and $H_i = (V_i, E_i)$ for $i \in \{1, \ldots, k\}$. Without loss of generality, assume that $\mathcal{H} \neq \emptyset$. First, a general framework is given for deriving approximation algorithms, which will be utilized in the rest of the section in multiple settings. The following definition plays a central role:

Definition 19 *Given an instance of the simultaneous assignment problem, we call m not necessarily distinct subsets $F_1, \ldots, F_m$ of the edges an* (m, ℓ)-cover *if every edge of G is contained in at least ℓ of $F_1, \ldots, F_m$.*

Theorem 20 *Given a linear program whose integer solutions are exactly the feasible simultaneous assignments, let $F_1, \ldots, F_m$ be an (m, ℓ)-cover of $G = (V, E)$ such that the polytope defined by the linear program becomes integer when restricted to F_i for all $i \in \{1, \ldots, m\}$. Then, the integrality gap of the linear program is at most $\frac{m}{\ell}$.*

Proof: Let x be an optimal solution to the linear program and let z be an optimal integer solution. Furthermore, let x_i and z_i denote an optimal fractional and integer solution to the problem restricted to F_i for $i \in \{1, \ldots, m\}$. Note that these solutions are also feasible solutions to the original problem. The following computation shows that the integrality gap is at most $\frac{m}{\ell}$.

$$\ell w x \leq \sum_{i=1}^{m} \sum_{e \in F_i} w(e)x(e) \leq \sum_{i=1}^{m} \sum_{e \in F_i} w(e)x_i(e) \leq m \sum_{e \in F_{i^*}} w(e)x_{i^*}(e) = m \sum_{e \in F_{i^*}} w(e)z_{i^*}(e) \leq mwz, \quad (9)$$

where $i^* = \arg\max_{i \in \{1,\ldots,m\}} \{\sum_{e \in F_i} w(e)x_i(e)\}$. The first inequality holds because every edge of G is contained in at least ℓ of $F_1, \ldots, F_m$, the second one follows by the optimality of x_i for F_i, the third one by the selection of i^*, whereas the equation holds because the polyhedron defined by the linear program is integer when the problem is restricted to F_{i^*}. By (9), one gets that $\frac{wx}{wz} \leq \frac{m}{\ell}$, which completes the proof. □

Theorem 20 gives a general framework for deriving bounds on the integrality gap. Note that the proof of Theorem 20 can be turned into an approximation algorithm if an (m, ℓ)-cover $F_1, \ldots, F_m$ is given and the linear program can be solved efficiently when the problem is restricted to F_i for all $i \in \{1, \ldots, m\}$ and m is polynomial in the size of the problem. In fact, one can avoid linear programming altogether and obtain an efficient $\frac{m}{\ell}$-approximation algorithm provided that the problems restricted to F_i are tractable and the heaviest among them can be found in polynomial time.

Theorem 21 *Let $F_1, \ldots, F_m$ be an (m, ℓ)-cover. Then, the heaviest among the optimal simultaneous assignments in the problems restricted to F_i is an $\frac{m}{\ell}$-approximate solution for the original problem.*

Proof: Let M_i denote an optimal solution to the problem restricted to F_i and let M^* be an optimal simultaneous assignment in G. Then,

$$\ell w(M^*) \leq \sum_{i=1}^{m} \sum_{e \in F_i \cap M^*} w(e) \leq \sum_{i=1}^{m} \sum_{e \in M_i} w(e) \leq m \sum_{e \in M_{i^*}} w(e) = mw(M_{i^*}) \quad (10)$$

holds, where $i^* = \arg\max\{w(M_i) : i \in \{1, \ldots, m\}\}$. This means that $\frac{w(M^*)}{w(M_{i^*})} \leq \frac{m}{\ell}$, that is, M_{i^*} is indeed $\frac{m}{\ell}$-approximate. Finally, observe that M_{i^*} is a feasible solution to the original problem. □

Theorem 21 gives a framework for deriving approximation algorithms for the simultaneous assignment problem. Namely, we need to find an (m, ℓ)-cover $F_1, \ldots, F_m$ — trying to minimize the ratio $\frac{m}{\ell}$ — such that one can find the best among the optimal solutions to the problems restricted to F_i for $i \in \{1, \ldots, m\}$, which is an $\frac{m}{\ell}$-approximate solution by Theorem 21. In fact, this framework easily extends to problems other than the simultaneous assignment problem — the main requirement is that any subset of a feasible solution should be feasible.

4.1 Approximation Algorithm for Uniform b

This section investigates the case of *uniform* b, that is, when there exists $a \in \mathbb{Z}_+$ such that $b_H \equiv a$ for all $H \in \mathcal{H}$. The following approach is an application of the approximation framework given above.

Theorem 22 *If b is uniform, then the integrality gap of (LP1*) is at most $\frac{k+1}{2}$, and a $\frac{k+1}{2}$-approximate solution can be found in strongly polynomial time.*

PROOF: Let x_i denote an optimal solution to the problem restricted to those edges which are in either $H_i \in \mathcal{H}$ or non of the subgraphs in $\mathcal{H}$, and let z denote an optimal solution to the problem restricted to those edges of G which are included in at most one of the subgraphs in $\mathcal{H}$. First, we show that (LP1*) defines an integer polyhedron for the problem restricted to H_i and also that x_i can be found in strongly polynomial time. In the problem restricted to H_i, all degree constraints posed in other subgraphs in $\mathcal{H}$ are redundant, hence we can delete all subgraphs other than H_i. The size of $\mathcal{H}$ being one, we conclude that x_i can be found in strongly polynomial time and (LP1*) defines an integer polyhedron for the restricted problem by Theorems 6 and 11. Second, if one restricts the problem to the edges contained in at most one of the subgraphs in $\mathcal{H}$, then the problem is again laminar. Therefore, z can be found in strongly polynomial time and the polytope defined by (LP1*) is integer when the problem is restricted to these edges.

Furthermore, observe that $z, x_1, \ldots, x_k$ is a $(k+1, 2)$-cover of E. By Theorem 20, this implies that the integrality gap of (LP1*) is at most $\frac{k+1}{2}$ and, by Theorem 21, the heaviest among $z, x_1, \ldots, x_k$ is a $\frac{k+1}{2}$-approximate solution, which completes the proof of the theorem. □

As a special case, this gives a $\frac{3}{2}$-approximation algorithm for the weighted double matching problem.

4.2 Approximation Algorithm for Small k

This section applies the approximation framework for the simultaneous assignment problem in the case when the size of $\mathcal{H}$ is small. Throughout this section, let $k = |\mathcal{H}|$ and let $k' \in \{1, \ldots, k\}$ be the smallest integer for which every edge appears in at most k' of the subgraphs in $\mathcal{H}$. Consider the following type of (m, ℓ)-covers.

Definition 23 *An (m, ℓ)-cover $F_1, \ldots, F_m$ is* laminar *if the problem restricted to F_i is laminar for each $i \in \{1, \ldots, m\}$.*

In the light of Theorems 20 and 21, we want to construct a laminar (m, ℓ)-cover minimizing the value $\frac{m}{\ell}$. Let $\alpha(k, k')$ denote the minimum value of $\frac{m}{\ell}$ for which a laminar (m', ℓ')-cover always exists such that $\frac{m'}{\ell'} \leq \frac{m}{\ell}$ whenever $k = |\mathcal{H}|$ and every edge appears in at most k' subgraphs in $\mathcal{H}$. In other words, $\alpha(k, k')$ is the best approximation ratio one can hope for by applying Theorem 21 to a laminar cover. The following min-max theorem gives an easy-to-compute formula for $\alpha(k, k')$.

Theorem 24 *Let k and k' be as above. The minimum value of $\frac{m}{\ell}$ for which there always exists a laminar (m', ℓ')-cover such that $\frac{m'}{\ell'} \leq \frac{m}{\ell}$, that is, $\alpha(k, k')$, equals*

$$\max_{j \in \{0, \ldots, k'-1\}} \frac{1}{k-j} \sum_{i=j+1}^{k'} \binom{k}{i}. \tag{11}$$

Furthermore, an $\alpha(k, k')$-approximate solution can be found in $\mathcal{O}(f(k)\,\mathrm{poly}(|V|, |E|))$ steps.

One possible approach to prove this is by a non-trivial reduction to the Duality theorem. For details, the reader is referred to [9].

Table 1 summarizes the value of $\alpha(k, k')$ given by Theorem 24 for small values of k and k'. By Theorems 21 and 24 we get the following.

k \ k'	1	2	3	4	5
1	1				
2	1	$3/2$			
3	1	2	$7/3$		
4	1	$5/2$	$7/2$	$15/4$	
5	1	3	5	$25/4$	$13/2$

Table 1: The approximation guarantees given by Theorem 24, where $k = |\mathcal{H}|$ and every edge is in at most k' subgraphs in $\mathcal{H}$. The highlighted values match the integrality gap of (LP1*).

Theorem 25 *One can find an $\alpha(k, k')$-approximate solution to the simultaneous assignment problem in $\mathcal{O}(f(k)\,\mathrm{poly}(|V|, |E|))$ time, where $k = |\mathcal{H}|$ and every edge appears in at most k' of the subgraphs in $\mathcal{H}$.*

For small k, this approximation guarantee is significantly better than that of the greedy algorithm, which is $(2k + 1)$.

By Theorem 11, (LP1*) defines an integer polyhedron when the problem is restricted to a locally laminar edge set, hence applying Theorems 20 and 24, one gets the following.

Theorem 26 *The integrality gap of (LP1*) is at most $\alpha(k, k')$, where $k = |\mathcal{H}|$ and every edge appears in at most k' of the subgraphs in $\mathcal{H}$.*

References

[1] R. P. Anstee. A polynomial algorithm for b-matchings: an alternative approach. *Information Processing Letters*, 24(3):153–157, 1987.

[2] K. Bérczi, A. Berger, M. Mnich, and R. Vincze. Degree-bounded generalized polymatroids and approximating the metric many-visits TSP. *arXiv preprint arXiv:1911.09890*, 2019.

[3] Y. Emek, S. Kutten, M. Shalom, and S. Zaks. Hierarchical b-matching. In *SOFSEM 2021: Theory and Practice of Computer Science: 47th International Conference on Current Trends in Theory and Practice of Computer Science, SOFSEM 2021, Bolzano-Bozen, Italy, January 25–29, 2021, Proceedings 47*, pages 189–202. Springer, 2021.

[4] A. Frank. *Connections in Combinatorial Optimization*. Oxford University Press, 2011.

[5] C. C. Huang and J. Ward. FPT-algorithms for the ℓ-matchoid problem with a coverage objective. *arXiv preprint arXiv:2011.06268*, 2020.

[6] T. A. Jenkyns. Matchoids: A generalization of matchings and matroids. *PhD thesis, University of Waterloo, Waterloo, Ontario*, 1974.

[7] K. Kaparis. On laminar matroids and b-matchings. *submitted for publication*, 2014.

[8] P. Madarasi. Matchings under distance constraints I. *Annals of Operations Research*, 305(1):137–161, 2021.

[9] P. Madarasi. The simultaneous assignment problem. *arXiv preprint arXiv:2105.09439*, 2021.

[10] J. Mestre. Greedy in approximation algorithms. In Yossi Azar and Thomas Erlebach, editors, *Algorithms – ESA 2006*, pages 528–539, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[11] A. Schrijver. *Combinatorial optimization: polyhedra and efficiency*, volume B. Springer, 2003.

Finding a PROPavg Allocation in Polynomial Time

Yusuke Kobayashi

Research Institute for Mathematical Sciences
Kyoto University, Japan
yusuke@kurims.kyoto-u.ac.jp

Ryoga Mahara

Research Institute for Mathematical Sciences
Kyoto University, Japan
ryoga@kurims.kyoto-u.ac.jp

Abstract: We study the problem of fairly allocating a set of indivisible goods to multiple agents and focus on the proportionality, which is one of the classical fairness notions. Since proportional allocations do not always exist when goods are indivisible, approximate concepts of proportionality have been considered in the previous work. Among them, proportionality up to the minimum valued good on average (PROPavg) has been the best approximate notion of proportionality that can be achieved for all instances. In this paper, we show that a PROPavg allocation can be computed in polynomial time. Our results establish PROPavg as a notable non-trivial fairness notion that can be achieved for all instances in polynomial time. Our algorithm is based on a generalized cut-and-choose protocol and a recursive technique.

Keywords: discrete fair division, indivisible goods, proportionality

1 Introduction

1.1 Proportional Allocation of Indivisible Goods

We study the problem of fairly allocating a set of indivisible goods to multiple agents under additive valuations. Fair division of indivisible goods is a fundamental and well-studied problem in Economics and Computer Science. We are given a set M of m indivisible goods and a set N of n agents with individual valuations. Under additive valuations, each agent $i \in N$ has value $v_i(\{g\}) \geq 0$ for each good g and her value for a bundle S of goods is equal to the sum of the value of each good $g \in S$, i.e., $v_i(S) = \sum_{g \in S} v_i(\{g\})$. An indivisible good can not be split among multiple agents and this causes finding a fair division to be a difficult task.

One of the standard notions of fairness is *proportionality*. Let $X = (X_1, X_2, \ldots, X_n)$ be an allocation, i.e., a partition of M into n bundles such that X_i is allocated to agent i. An allocation X is said to be *proportional (PROP)* if $v_i(X_i) \geq \frac{1}{n} v_i(M)$ holds for each agent i. In other words, in a proportional allocation, every agent receives a set of goods whose value is at least $1/n$ fraction of the value of the entire set. Unfortunately, proportional allocations do not always exist when goods are indivisible. For instance, when allocating a single indivisible good to more than one agents it is impossible to achieve any proportional allocation. Thus, several relaxations of proportionality such as PROP1, PROPx, and PROPm have been considered in the previous work.

Each of these notions requires that each agent $i \in N$ receives value at least $\frac{1}{n} v_i(M) - d_i(X)$, where $d_i(X)$ is appropriately defined for each notion. *Proportionality up to the largest valued good (PROP1)* is a relaxation of proportionality that was introduced by Conitzer et al. [17]. PROP1 requires $d_i(X)$ to be the largest value that agent i has for any good allocated to other agents, i.e., $d_i(X) = \max_{k \in N \setminus \{i\}} \max_{g \in X_k} v_i(\{g\})$. It is shown in [17] that there always exists a Pareto optimal[1] allocation that satisfies PROP1. Moreover, Aziz et al. [4] presented a polynomial-time algorithm that finds a PROP1 and Pareto optimal allocation even in the presence of chores, i.e., some items can have negative value.

[1] An allocation $X = (X_1, \ldots, X_n)$ is *Pareto optimal* if there is no allocation $Y = (Y_1, \ldots, Y_n)$ such that $v_i(Y_i) \geq v_i(X_i)$ for any agent i, and there exists an agent j such that $v_j(Y_j) > v_j(X_j)$.

Another relaxation is *proportionality up to the least valued good (PROPx)*, which is much stronger than PROP1. PROPx requires $d_i(X)$ to be the least value that agent i has for any good allocated to other agents, i.e., $d_i(X) = \min_{k \in N \setminus \{i\}} \min_{g \in X_k} v_i(\{g\})$. Moulin [26] gave an example for which no PROPx allocation exists, and Aziz et al. [4] gave a simpler example.

Recently, Baklanov et al. [5] introduced *proportionality up to the maximin good (PROPm)*. PROPm requires $d_i(X) = \max_{k \in N \setminus \{i\}} \min_{g \in X_k} v_i(\{g\})$, which shows that PROPm is the notion between PROP1 and PROPx. It is shown in [5] that a PROPm allocation always exists for instances with at most five agents, and later Baklanov et al. [6] showed that there always exists a PROPm allocation for any instance and it can be computed in polynomial time.

However, in some cases, PROPm is not a good enough relaxation of proportionality. Suppose that there exists a good $g \in M$ for which every agent has value at least $1/n$ fraction of the value of M. Then allocating g to some agent i and allocating all the goods in $M \setminus \{g\}$ to another agent achieves a PROPm allocation, whereas it will be better to allocate $M \setminus \{g\}$ to $N \setminus \{i\}$ in a fair manner. This motivates the study of better relaxations of proportionality than PROPm. Very recently, Kobayashi and Mahara [21] introduced *proportionality up to the least valued good on average (PROPavg)*, a new relaxation of proportionality, and showed that a PROPavg allocation always exists for all instances.

1.2 Our Contribution

In this paper, we show that a PROPavg allocation can be computed in polynomial time. PROPavg requires $d_i(X)$ to be the average of minimum value that agent i has for any good allocated to other agents, i.e., $d_i(X) = \frac{1}{n-1} \sum_{k \in N \setminus \{i\}} \min_{g \in X_k} v_i(\{g\})$. It is easy to see that PROPavg implies PROPm. Note that a similar and slightly stronger notion than PROPavg was introduced by Baklanov et al. [5] with the name of Average-EFX (Avg-EFX), where $d_i(X) = \frac{1}{n} \sum_{k \in N \setminus \{i\}} \min_{g \in X_k} v_i(\{g\})$. It remains open whether an Avg-EFX allocation always exists. The main contribution of this paper is to show the following theorem which extends the results on PROPm allocations shown by Baklanov et al. [6].

Theorem 1 *A PROPavg allocation can be computed in polynomial time when each agent has a non-negative additive valuation.*

1.3 Related Work

Fair division of divisible resources is a classical topic starting from the 1940's [29] and has a long history in multiple fields such as Economics, Social Choice Theory, and Computer Science [9,10,25,28]. In contrast, fair division of indivisible items has been actively studied in recent years (see, e.g., [2,3]).

In the context of fair division, besides proportionality, *envy-freeness* is another well-studied notion of fairness. An allocation is called *envy-free (EF)* if each agent receives a set of goods that she values at least as much as any other agent's goods. As in the proportionality case, envy-free allocations do not always exist when goods are indivisible, and several relaxations of envy-freeness have been considered. Among them, a notable one is *envy-freeness up to one good (EF1)* [11]. It is known that an EF1 allocation always exists, and it can be computed in polynomial time [22]. Another notable relaxation is *envy-freeness up to any good (EFX)* [13]. An allocation $X = (X_1, \ldots, X_n)$ is called EFX if for any pair of agents $i, j \in N$, $v_i(X_i) \geq v_i(X_j) - m_i(X_j)$, where $m_i(X_j)$ is the value of the least valued good for agent i in X_j. Whether EFX allocations always exist or not is one of the major open problems in fair division.

There have been several studies on the existence of an EFX allocation for restricted cases. Plaut and Roughgarden [27] showed that an EFX allocation always exists for instances with two agents even when each agent can have more general valuations than additive valuations. Chaudhury et al. [14] showed that an EFX allocation always exists for instances with three agents. It is not known whether EFX allocations always exist even for instances with four agents having additive valuations. As mentioned in [5], it is easy to see that EFX implies Avg-EFX. As with EFX, whether Avg-EFX allocations always exist is not known even for instances with four or more agents. We can also consider the cases with restricted valuations. For example, there always exists an EFX allocation when valuations are identical [27], two types [23,24], binary [7,18], or bi-valued [1].

Another direction of research related to EFX is *EFX-with-charity*, in which unallocated goods are allowed. Obviously, without any constraints, the problem is trivial: leaving all goods unallocated results in an envy-free allocation. Thus, the goal here is to find allocations with better guarantees. For additive valuations, Caragiannis et al. [12] showed that there exists an EFX allocation with some unallocated goods where every agent receives at least half the value of her bundle in a maximum *Nash social welfare* allocation[2]. For normalized and monotone valuations, Chaudhury et al. [16] showed that there exist an EFX allocation and a set of unallocated goods U such that every agent has value for her own bundle at least her value for U, and $|U| < n$. Berger et al. [8] showed that the number of the unallocated goods can be decreased to $n-2$, and to just one for the case of four agents having nice cancelable valuations, which are more general than additive valuations. Mahara [24] showed that the number of the unallocated goods can be decreased to $n-2$ for normalized and monotone valuations, which are more general than nice cancelable valuations. For additive valuations, Chaudhury et al. [15] presented a polynomial-time algorithm for finding an approximate EFX allocation with at most a sublinear number of unallocated goods and high Nash social welfare.

2 Preliminaries

Let $N = \{1, \ldots, n\}$ be a set of n agents and M be a set of m goods. We assume that goods are indivisible: a good can not be split among multiple agents. Each agent $i \in N$ has a non-negative valuation $v_i : 2^M \to \mathbb{R}_{\geq 0}$, where 2^M is the power set of M. We assume that each valuation v_i is *normalized*: $v_i(\emptyset) = 0$, *monotone*: $S \subseteq T$ implies $v_i(S) \leq v_i(T)$ for any $S, T \subseteq M$, and *additive*: $v_i(S) = \sum_{g \in S} v_i(\{g\})$ for any $S \subseteq M$. For ease of explanation, we normalize the valuations so that $v_i(M) = 1$ for all $i \in N$.

To simplify notation, we denote $\{1, \ldots, k\}$ by $[k]$ for any positive integer k, write $v_i(g)$ instead of $v_i(\{g\})$ for $g \in M$, and use $S \setminus g$ and $S \cup g$ instead of $S \setminus \{g\}$ and $S \cup \{g\}$, respectively.

We say that $X = (X_1, X_2, \ldots, X_n)$ is an *allocation of M to N* if it is a partition of M into n disjoint subsets such that each set is indexed by $i \in N$. Each X_i is the set of goods given to agent i, which we call a *bundle*. It is simply called an *allocation to N* if M is clear from context. For $i \in N$ and $S \subseteq M$, let $m_i(S)$ denote the value of the least valuable good for agent i in S, that is, $m_i(S) = \min_{g \in S}\{v_i(g)\}$ if $S \neq \emptyset$ and $m_i(\emptyset) = 0$. For an allocation $X = (X_1, X_2, \ldots, X_n)$ to N, we say that an agent i is *PROPavg-satisfied* by X if

$$v_i(X_i) + \frac{1}{n-1} \sum_{k \in [n] \setminus i} m_i(X_k) \geq \frac{1}{n},$$

where we recall that $v_i(M) = 1$. In other words, agent i receives a set of goods for which she has value at least $1/n$ fraction of her total value minus the average of minimum value of the set of goods any other agent receives. An allocation X is called *PROPavg* if every agent $i \in N$ is PROPavg-satisfied by X.

Let $G = (V, E)$ be an undirected graph. For $v \in V$, let $G - v$ denote the graph obtained from G by deleting v. A *perfect matching* in G is a set of pairwise disjoint edges of G covering all the vertices of G.

3 PROPavg-Graph

In order to prove Theorem 1, we give an algorithm for finding a PROPavg allocation by improving the previous one in [21]. Let us briefly explain the previous algorithm in [21]. This algorithm is a generalization of the cut-and-choose protocol that consists of the following three steps.

1. We partition the goods into n bundles without assigning them to agents.
2. A specified agent, say n, chooses the best bundle for her valuation.
3. We determine an assignment of the remaining bundles to the agents in $N \setminus n$.

[2] This is an allocation that maximizes $\Pi_{i=1}^{n} v_i(X_i)$.

The partition given in the first step is represented by an allocation of M to a newly introduced set of size n, say V_2, and the assignment in the third step is represented by a matching in an auxiliary bipartite graph, which we call *PROPavg-graph*. In this section, we define the PROPavg-graph and its desired properties.

Let V_2 be a set of n elements and fix a specified element $r \in V_2$. We say that $X = (X_u)_{u \in V_2}$ is an *allocation to* V_2 if it is a partition of M into n disjoint subsets such that each set is indexed by an element in V_2, that is, $\bigcup_{u \in V_2} X_u = M$ and $X_u \cap X_{u'} = \emptyset$ for distinct $u, u' \in V_2$. For an allocation $X = (X_u)_{u \in V_2}$ to V_2, we define a bipartite graph $G_X = (V_1, V_2; E)$ called *PROPavg-graph* as follows. The vertex set consists of $V_1 = N \setminus n$ and V_2, and the edge set E is defined by

$$(i, u) \in E \iff v_i(X_u) + \frac{1}{n-1} \sum_{u' \in V_2 \setminus \{r,u\}} m_i(X_{u'}) \geq \frac{1}{n}$$

for $i \in V_1$ and $u \in V_2$. It should be emphasized that the summation is taken over $V_2 \setminus \{r, u\}$, i.e., $m_i(X_r)$ is not counted, in the above definition. The following lemma shows that the PROPavg-graph is closely related to the definition of PROPavg-satisfaction.

Lemma 2 (Kobayashi and Mahara [21]) *Suppose that $G_X = (V_1, V_2; E)$ is the PROPavg-graph for an allocation $X = (X_u)_{u \in V_2}$ to V_2. Let σ be a bijection from N to V_2 and define an allocation $Y = (Y_1, \ldots, Y_n)$ to N by $Y_i = X_{\sigma(i)}$ for $i \in N$. For $i^* \in V_1$, if $(i^*, \sigma(i^*)) \in E$, then i^* is PROPavg-satisfied by Y.*

As we will see in Section 4, throughout the algorithm in [21], we always keep an allocation $X = (X_u)_{u \in V_2}$ to V_2 that satisfies the following property.

(P1) $G_X - r$ has a perfect matching.

By updating allocation X repeatedly while keeping (P1), we construct an allocation that satisfies the following stronger property.

(P2) For any $u \in V_2$, $G_X - u$ has a perfect matching.

4 Existence of a PROPavg Allocation

In this section, we briefly show the pseudo-polynomial algorithm in [21] to find a PROPavg allocation. The algorithm begins with obtaining an initial allocation $X = (X_u)_{u \in V_2}$ to V_2 satisfying (P1). Unless X satisfies (P2), we appropriately choose a good in $\bigcup_{u \in V_2 \setminus r} X_u$ and move it to X_r while keeping (P1). Finally, we get an allocation $X^* = (X^*_u)_{u \in V_2}$ to V_2 satisfying (P2). As we will see in Lemma 4, we can obtain a PROPavg allocation to N by using this allocation.

Lemma 3 (Kobayashi and Mahara [21]) *There exists an allocation $X = (X_u)_{u \in V_2}$ to V_2 satisfying (P1).*

The following lemma shows that if we obtain an allocation $X = (X_u)_{u \in V_2}$ to V_2 satisfying (P2), then we can find a PROPavg allocation to N in polynomial time.

Lemma 4 *Suppose that $X = (X_u)_{u \in V_2}$ is an allocation to V_2 satisfying (P2). Then, we can construct a PROPavg allocation to N in polynomial time.*

Proof: Let $X = (X_u)_{u \in V_2}$ be an allocation to V_2 satisfying (P2). First, agent n chooses the best bundle X_{u^*} for her valuation among $\{X_u \mid u \in V_2\}$ (if there is more than one such bundle, choose one arbitrarily). Since X satisfies (P2), there exists a perfect matching A in $G_X - u^*$. For each agent $i \in V_1 (= N \setminus n)$, the bundle corresponding to the vertex that matches i in A is allocated to i. By Lemma 2, i is PROPavg-satisfied for each agent $i \in V_1$. Furthermore, since we have $v_n(X_{u^*}) = \max_{u \in V_2} v_n(X_u) \geq \frac{1}{n}$, agent n is

also PROPavg-satisfied. Therefore, the obtained allocation is a PROPavg allocation. Furthermore, such an allocation can be found in polynomial-time by a maximum matching algorithm. □

The following proposition shows how to update an allocation in each iteration.

Proposition 5 (Kobayashi and Mahara [21]) *Suppose that $X = (X_u)_{u \in V_2}$ is an allocation to V_2 that satisfies (P1) but does not satisfy (P2). Then, there exists another allocation $X' = (X'_u)_{u \in V_2}$ to V_2 satisfying (P1) such that $|X'_r| = |X_r| + 1$.*

We note that the allocation X' in Proposition 5 is obtained by moving an appropriate item $g \in \bigcup_{u \in V_2 \setminus r} X_u$ to X_r.

In summary, the algorithm in [21] find a PROPavg allocation as follows. See Algorithm 1 for the algorithm description. By Lemma 3, we first obtain an initial allocation $X = (X_u)_{u \in V_2}$ to V_2 satisfying (P1). By Proposition 5, unless X satisfies (P2), we can increase $|X_r|$ by one while keeping the property (P1). Since $|X_r| \leq |M|$, this procedure terminates in at most m steps, and we finally obtain an allocation X^* to V_2 satisfying (P2). Therefore, there exists a PROPavg allocation to N by Lemma 4.

Algorithm 1 Algorithm for finding a PROPavg allocation

Input: agents N, goods M, and a valuation v_i for each $i \in N$
Output: a PROPavg allocation to N

1: Apply Lemma 3 to obtain an allocation X to V_2 satisfying (P1).
2: **while** X does not satisfy (P2) **do**
3: Apply Proposition 5 to X and obtain another allocation X' to V_2.
4: $X \leftarrow X'$.
5: Apply Lemma 4 to obtain a PROPavg allocation to N.

Algorithm 1 runs in pseudo-polynomial time. This is because we use the algorithm in [16] as a subroutine in order to obtain an initial allocation X to V_2 satisfying (P1). Actually, the algorithm in [16] only leads to a pseudo-polynomial time algorithm when each valuations are additive. We give a polynomial-time algorithm to find a PROPavg allocation by improving Algorithm 1 in Section 5.

5 Finding a PROPavg Allocation in Polynomial Time

In this section, we show how to find a PROPavg allocation in polynomial time. As mentioned in Section 4, Algorithm 1 runs in pseudo-polynomial time. This is because we can not guarantee the polynomial solvability in line 1 of Algorithm 1. We can see that the other parts of Algorithm 1 run in polynomial time as follows. In line 2, we can check (P2) in polynomial time by applying a maximum matching algorithm for each $G_X - u$. In line 3, it suffices to find a good $g \in \bigcup_{u \in V_2 \setminus r} X_u$ such that (P1) is kept after moving g. Since (P1) can be checked in polynomial time, this can be done in polynomial time by considering all g in a brute-force way. Finally, line 5 is executed in polynomial time by Lemma 4. Note that we can speed up lines 2 and 3 by using the DM-decomposition of G_X [19,20], but we do not go into details, because we only focus on the polynomial solvability.

Let us now consider how to find an initial allocation X to V_2 satisfying (P1) in polynomial time. Our idea is to use a recursive algorithm. That is, we use a PROPavg allocation of M to $n-1$ agents as an initial allocation X to V_2 satisfying (P1). Indeed, if it holds that $v_i(g) \leq \frac{1}{n}$ for any agent $i \in N$ and any good $g \in M$, then we can show that a PROPavg allocation of M to $n-1$ agents satisfies (P1) as follows.

Lemma 6 *Suppose that for any agent $i \in N$ and any good $g \in M$, we have $v_i(g) \leq \frac{1}{n}$. Let $(X_1, \ldots, X_{n-1})$ be a PROPavg allocation for $N \setminus n$. Then, $X = (X_1, \ldots, X_{n-1}, X_n)$ is an allocation to $V_2 = [n]$ satisfying (P1), where $X_n = \emptyset$ and the specific element $r \in V_2$ is equal to n.*

PROOF: Let $G_X = (V_1, V_2; E)$ be the PROPavg-graph corresponding to X. It is enough to show that $(i, X_i) \in E$ for any $i \in [n-1]$. Fix any $i \in [n-1]$. We obtain that

$$\begin{aligned} v_i(X_i) &\geq \frac{1}{n-1} - \frac{1}{n-2} \sum_{j \in [n-1] \setminus i} m_i(X_j) \\ &= \frac{1}{n} - \frac{1}{n-1} \sum_{j \in [n-1] \setminus i} m_i(X_j) \\ &\quad + \underbrace{\frac{1}{n-1} \left(\frac{1}{n} - \frac{1}{n-2} \sum_{j \in [n-1] \setminus i} m_i(X_j) \right)}_{\geq 0} \\ &\geq \frac{1}{n} - \frac{1}{n-1} \sum_{j \in [n-1] \setminus i} m_i(X_j), \end{aligned}$$

where the first inequality follows from the assumption that $(X_1, \ldots, X_{n-1})$ is a PROPavg allocation and the second inequality follows from the assumption that $v_i(g) \leq \frac{1}{n}$ for any $i \in N$ and $g \in M$. This implies that $(i, X_i) \in E$ and thus X is an allocation to $V_2 = [n]$ satisfying (P1). □

Unfortunately, the argument in Lemma 6 does not work without the assumption that $v_i(g) \leq \frac{1}{n}$ for any $i \in N$ and $g \in M$. To elude this difficulty, our algorithm applies preprocessing. This preprocessing allocates g to i and remove i and g from our instance as long as there exists an agent i and a good g such that $v_i(g) \leq \frac{1}{n}$. See Algorithm 2 for the entire algorithm.

If this preprocessing removes at least one agent from our instance, then our algorithm recursively computes a PROPavg allocation for the remaining agents and goods, and return the overall allocation together with the removed agents. In order to verify that the returned allocation is a PROPavg allocation for n agents, we need a refined inequality (see line 7 of Algorithm 2).

Otherwise, our algorithm recursively computes a PROPavg allocation for $n-1$ agents. Since $v_i(g) < \frac{1}{n}$ holds for any agent i and good g, we can use this allocation as an initial allocation to V_2 satisfying (P1) by Lemma 6. The rest of our algorithm finds an allocation to V_2 satisfying (P2) and return a PROPavg allocation as in Algorithm 1.

In the remaining part of this section, we prove the correctness of Algorithm 2 and the latter part of Theorem 1. The following lemma shows that if the preprocessing removes at least one agent from our instance, then algorithm returns a legal PROPavg allocation for N.

Lemma 7 *In line 14 of Algorithm 2, $X = (X_1, \ldots, X_{|N|})$ is a PROPavg allocaiton to N.*

PROOF: Fix any $i \in N$. We show that i is PROPavg-satisfied by X.

Case 1: $i \in N_2$

In this case, agent i receives exactly one good in the while statement. By the while condition, we have

$$\begin{aligned} v_i(X_i) &\geq \frac{1}{n} - \frac{1}{n-1} \sum_{j \in N_2} m_i(X_j) \\ &\geq \frac{1}{n} - \frac{1}{n-1} \sum_{j \in N \setminus i} m_i(X_j). \end{aligned}$$

Thus, i is PROPavg-satisfied by X.

Algorithm 2 Algorithm for finding a PROPavg allocation in polynomial time

procedure PROPAVG(N, M, $\{v_i\}_{i\in N}$)
  if $|N| = 1$ **then**
    return $X = (M)$
  else
    $N_1 \leftarrow N, N_2 \leftarrow \emptyset$
    $M_1 \leftarrow M, M_2 \leftarrow \emptyset$
    while $\exists i \in N_1$ and $\exists g \in M_1$ s.t. $v_i(g) \geq \frac{v_i(M)}{|N|} - \frac{1}{|N|-1}\sum_{j\in N_2} m_i(X_j)$ **do**
      $X_i \leftarrow \{g\}$
      $N_1 \leftarrow N_1 \setminus i, N_2 \leftarrow N_2 \cup i$
      $M_1 \leftarrow M_1 \setminus g, M_2 \leftarrow M_2 \cup g$
    Let $N_1 = \{1, \ldots, l\}$ and $N_2 = \{l+1, \ldots, |N|\}$, renumbering if necessary.
    if $|N_2| \geq 1$ **then**
      $(X_1, \ldots, X_l) \leftarrow$ PROPAVG(N_1, M_1, $\{v_i\}_{i\in N_1}$)
      return $X = (X_1, \ldots, X_{|N|})$
    else
      $(X_1, \ldots, X_{n-1}) \leftarrow$ PROPAVG($N \setminus n$, M, $\{v_i\}_{i\in N\setminus n}$) ▷ $N_1 = N, M_1 = M$
      Apply Lemma 6 to obtain an allocation $X = (X_1, \ldots, X_n)$ satisfying (P1).
      while X does not satisfy (P2) **do**
        Apply Proposition 5 to X and obtain another allocation X' to V_2.
        $X \leftarrow X'$.
      Apply Lemma 4 to obtain a PROPavg allocation $X = (X_1, \ldots, X_{|N|})$to N.
      return $X = (X_1, \ldots, X_{|N|})$

Case 2: $i \in N_1$ and $l = 1$

In this case, we have

$$\begin{aligned} v_i(X_i) = v_i(M_1) &= v_i(M) - \sum_{j\in N_2} m_i(X_j) \\ &= \left(\frac{1}{n} - \frac{1}{n-1}\sum_{j\in N_2} m_i(X_j)\right) + \underbrace{\left(\frac{n-1}{n} - \frac{n-2}{n-1}\sum_{j\in N_2} m_i(X_j)\right)}_{\geq 0} \\ &\geq \frac{1}{n} - \frac{1}{n-1}\sum_{j\in N_2} m_i(X_j) \\ &= \frac{1}{n} - \frac{1}{n-1}\sum_{j\in N\setminus i} m_i(X_j), \end{aligned}$$

where the last inequality follows from $\frac{n-1}{n} \geq \frac{n-2}{n-1} \geq \frac{n-2}{n-1}\sum_{j\in N_2} m_i(X_j)$.

Case 3: $i \in N_1$ and $l \geq 2$

Since $(X_1, \ldots, X_l)$ is a PROPavg allocation of M_1 to N_1, we have

$$\begin{aligned} v_i(X_i) &\geq \frac{v_i(M_1)}{l} - \frac{1}{l-1}\sum_{j\in N_1\setminus i} m_i(X_j) \\ &= \frac{1}{l} - \frac{1}{l}\sum_{j\in N_2} m_i(X_j) - \frac{1}{l-1}\sum_{j\in N_1\setminus i} m_i(X_j). \end{aligned} \tag{1}$$

In line 13 of Algorithm 2, the while condition in line 7 does not hold for any agent in N_1. Thus, it holds that

$$m_i(X_j) < \frac{1}{n} - \frac{1}{n-1} \sum_{j \in N_2} m_i(X_j) \tag{2}$$

for any $j \in N_1 \setminus i$. Summing up inequality (2) for each $j \in N_1 \setminus i$, we obtain

$$\sum_{j \in N_1 \setminus i} m_i(X_j) < \frac{l-1}{n} - \frac{l-1}{n-1} \sum_{j \in N_2} m_i(X_j). \tag{3}$$

By multiplying inequality (3) by $\frac{n-l}{l(l-1)}$ and rearranging, we have

$$0 > -\frac{n-l}{ln} + \frac{n-l}{l(n-1)} \sum_{j \in N_2} m_i(X_j) + \frac{n-l}{l(l-1)} \sum_{j \in N_1 \setminus i} m_i(X_j). \tag{4}$$

Summing up inequalities (1) and (4), we have

$$v_i(X_i) > \frac{1}{n} + \left(-\frac{1}{l} + \frac{n-l}{l(n-1)} \right) \sum_{j \in N_2} m_i(X_j) + \left(-\frac{1}{l-1} + \frac{n-l}{l(l-1)} \right) \sum_{j \in N_1 \setminus i} m_i(X_j). \tag{5}$$

By direct calculation, we have

$$-\frac{1}{l} + \frac{n-l}{l(n-1)} + \frac{1}{n-1} = \frac{1}{l(n-1)} \geq 0$$

and

$$\begin{aligned} -\frac{1}{l-1} + \frac{n-l}{l(l-1)} + \frac{1}{n-1} &= \frac{1}{l(l-1)(n-1)} \left(-l(n-1) + (n-l)(n-1) + l(l-1) \right) \\ &= \frac{(n-l)(n-l-1)}{l(l-1)(n-1)} \\ &\geq 0. \end{aligned}$$

Applying these inequalities to inequality (5), we finally obtain

$$\begin{aligned} v_i(X_i) &> \frac{1}{n} - \frac{1}{n-1} \left(\sum_{j \in N_2} m_i(X_j) + \sum_{j \in N_1 \setminus i} m_i(X_j) \right). \\ &= \frac{1}{n} - \frac{1}{n-1} \sum_{j \in N \setminus i} m_i(X_j), \end{aligned}$$

which implies that i is PROPavg-satisfied by X.

Therefore, X is a PROPavg allocation to N in line 16 of Algorithm 2. □

We finally give the proof of Theorem 1 by showing that Algorithm 2 is a polynomial time algorithm to find a PROPavg allocation.

Proof:[Proof of Theorem 1] We first show the correctness of Algorithm 2. If $|N| = 1$, our algorithm obviously returns a PROPavg allocation in line 3. Assume that $|N| \geq 2$. If $|N_2| \geq 1$, it returns a PROPavg allocation in line 14 by Lemma 7. Otherwise, since the while condition in line 7 does not hold for any agent in N_1, $v_i(g) < \frac{1}{n}$ holds for any agent $i \in N_1$ and good $g \in M_1$ in line 16. Thus, $X = (X_1, \ldots, X_n)$ satisfies (P1) by Lemma 6, where X_n is an empty set. The rest of the algorithm finds an allocation to

V_2 satisfying (P2) and return a PROPavg allocation as in Algorithm 1. Therefore, Algorithm 2 returns a PROPavg allocation in all cases.

We finally show that Algorithm 2 completes in time polynomial in the number of agents and items.

Let $T(n, m)$ be the worst case time complexity of Algorithm 2 when $|N| = n$ and $|M| = m$. Clearly, $T(1, m) = O(1)$. We can check the while condition in line 7 and execute the body of the while loop in polynomial time of n and m. In addition, as mentioned at the beginning of Section 5, Lines 17 to 22 can be executed in polynomial time of n and m. Thus, $T(n, m)$ can be expressed as

$$T(n,m) = \mathrm{poly}(n,m) + \max\{\max_{\substack{1\le n'\le n-1\\ 1\le m'\le m-1}} T(n',m'), T(n-1,m)\}$$

Therefore, $T(n, m)$ is polynomially bounded in n and m.

□

Acknowledgments

This work was partially supported by the joint project of Kyoto University and Toyota Motor Corporation, titled "Advanced Mathematical Science for Mobility Society", and by JSPS, KAKENHI grant number JP19H05485, Japan.

References

[1] Georgios Amanatidis, Georgios Birmpas, Aris Filos-Ratsikas, Alexandros Hollender, and Alexandros A Voudouris. Maximum Nash welfare and other stories about EFX. *Theoretical Computer Science*, 863:69–85, 2021.

[2] Georgios Amanatidis, Georgios Birmpas, Aris Filos-Ratsikas, and Alexandros A Voudouris. Fair division of indivisible goods: A survey. *arXiv preprint arXiv:2202.07551*, 2022.

[3] Haris Aziz, Bo Li, Herve Moulin, and Xiaowei Wu. Algorithmic fair allocation of indivisible items: A survey and new questions. *arXiv preprint arXiv:2202.08713*, 2022.

[4] Haris Aziz, Hervé Moulin, and Fedor Sandomirskiy. A polynomial-time algorithm for computing a Pareto optimal and almost proportional allocation. *Operations Research Letters*, 48(5):573–578, 2020.

[5] Artem Baklanov, Pranav Garimidi, Vasilis Gkatzelis, and Daniel Schoepflin. Achieving proportionality up to the maximin item with indivisible goods. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 5143–5150, 2021.

[6] Artem Baklanov, Pranav Garimidi, Vasilis Gkatzelis, and Daniel Schoepflin. PROPm allocations of indivisible goods to multiple agents. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 24–30, 2021.

[7] Siddharth Barman, Sanath Kumar Krishnamurthy, and Rohit Vaish. Greedy algorithms for maximizing Nash social welfare. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, pages 7–13, 2018.

[8] Ben Berger, Avi Cohen, Michal Feldman, and Amos Fiat. (Almost full) EFX exists for four agents (and beyond). *arXiv preprint arXiv:2102.10654*, 2021.

[9] Steven J Brams, Steven John Brams, and Alan D Taylor. *Fair Division: From cake-cutting to dispute resolution*. Cambridge University Press, 1996.

[10] Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D Procaccia. *Handbook of computational social choice*. Cambridge University Press, 2016.

[11] Eric Budish. The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes. *Journal of Political Economy*, 119(6):1061–1103, 2011.

[12] Ioannis Caragiannis, Nick Gravin, and Xin Huang. Envy-freeness up to any item with high Nash welfare: The virtue of donating items. In *Proceedings of the 20th ACM Conference on Economics and Computation (EC)*, pages 527–545, 2019.

[13] Ioannis Caragiannis, David Kurokawa, Hervé Moulin, Ariel D. Procaccia, Nisarg Shah, and Junxing Wang. The unreasonable fairness of maximum Nash welfare. *ACM Transactions on Economics and Computation*, 7(3):1–32, 2019.

[14] Bhaskar Ray Chaudhury, Jugal Garg, and Kurt Mehlhorn. EFX exists for three agents. In *Proceedings of the 21st ACM Conference on Economics and Computation (EC)*, pages 1–19, 2020.

[15] Bhaskar Ray Chaudhury, Jugal Garg, Kurt Mehlhorn, Ruta Mehta, and Pranabendu Misra. Improving EFX guarantees through rainbow cycle number. In *Proceedings of the 22nd ACM Conference on Economics and Computation (EC)*, pages 310–311, 2021.

[16] Bhaskar Ray Chaudhury, Telikepalli Kavitha, Kurt Mehlhorn, and Alkmini Sgouritsa. A little charity guarantees almost envy-freeness. *SIAM Journal on Computing*, 50(4):1336–1358, 2021.

[17] Vincent Conitzer, Rupert Freeman, and Nisarg Shah. Fair public decision making. In *Proceedings of the 18th ACM Conference on Economics and Computation (EC)*, pages 629–646, 2017.

[18] Andreas Darmann and Joachim Schauer. Maximizing Nash product social welfare in allocating indivisible goods. *European Journal of Operational Research*, 247(2):548–559, 2015.

[19] Andrew L Dulmage. A structure theory of bipartite graphs of finite exterior dimension. *The Transactions of the Royal Society of Canada, Section III*, 53:1–13, 1959.

[20] Andrew L Dulmage and Nathan S Mendelsohn. Coverings of bipartite graphs. *Canadian Journal of Mathematics*, 10:517–534, 1958.

[21] Yusuke Kobayashi and Ryoga Mahara. Proportional Allocation of Indivisible Goods up to the Least Valued Good on Average. In *33rd International Symposium on Algorithms and Computation (ISAAC 2022)*, pages 55:1–55:13, 2022.

[22] Richard J. Lipton, Evangelos Markakis, Elchanan Mossel, and Amin Saberi. On approximately fair allocations of indivisible goods. In *Proceedings of the 5th ACM Conference on Electronic Commerce (EC)*, pages 125–131, 2004.

[23] Ryoga Mahara. Existence of EFX for two additive valuations. *arXiv preprint arXiv:2008.08798*, 2020.

[24] Ryoga Mahara. Extension of additive valuations to general valuations on the existence of EFX. In *Proceedings of the 29th Annual European Symposium on Algorithms (ESA)*, pages 66:1–15, 2021.

[25] Hervé Moulin. *Fair division and collective welfare*. MIT press, 2004.

[26] Hervé Moulin. Fair division in the internet age. *Annual Review of Economics*, 11:407–441, 2019.

[27] Benjamin Plaut and Tim Roughgarden. Almost envy-freeness with general valuations. *SIAM Journal on Discrete Mathematics*, 34(2):1039–1068, 2020.

[28] Jack Robertson and William Webb. Cake-cutting algorithms: Be fair if you can, 1998.

[29] Hugo Steinhaus. The problem of fair division. *Econometrica*, 16(1):101–104, 1948.

Weighted exchange distance of basis pairs

Kristóf Bérczi[1]

MTA-ELTE Matroid Optimization Research Group
ELKH-ELTE Egerváry Research Group
Department of Operations Research
Eötvös Loránd University
Budapest, Hungary
kristof.berczi@ttk.elte.hu

Bence Mátravölgyi[1]

MTA-ELTE Matroid Optimization Research Group
Department of Operations Research
Eötvös Loránd University
Budapest, Hungary
matben@student.elte.hu

Tamás Schwarcz[12]

MTA-ELTE Matroid Optimization Research Group
Department of Operations Research
Eötvös Loránd University
Budapest, Hungary
tamas.schwarcz@ttk.elte.hu

Abstract: Two pairs of disjoint bases $\mathbf{P}_1 = (R_1, B_1)$ and $\mathbf{P}_2 = (R_2, B_2)$ of a matroid M are called *equivalent* if $\mathbf{P}_1$ can be transformed into $\mathbf{P}_2$ by a series of symmetric exchanges. In 1980, White conjectured that such a sequence always exists whenever $R_1 \cup B_1 = R_2 \cup B_2$. A strengthening of the conjecture was proposed by Hamidoune, stating that minimum length of an exchange is at most the rank of the matroid.

We propose a weighted variant of Hamidoune's conjecture, where the weight of an exchange depends on the weights of the exchanged elements. We prove the conjecture for several matroid classes: strongly base orderable matroids, split matroids, graphic matroids of wheels, and spikes.

Keywords: Graphic matroid, Sequential symmetric basis exchange, Spike, Split matroid, Strongly base orderable matroid, Wheel graph

1 Introduction

Given a matroid M over a ground set S, the exchange axiom implies that for any pair of bases R and B there exists a sequence of exchanges that transforms R into B, and the shortest length of such a sequence is $|R - B|$. In the light of this, it is natural to ask whether analogous results hold for basis pairs instead of single basis. More precisely, let (R, B) be an ordered pair of disjoint bases of M, and let $e \in R \setminus B$ and $f \in B \setminus R$ be such that both $R' \coloneqq R - e + f$ and $B' \coloneqq B + e - f$ are bases. In such a case, we call the exchange **feasible** and say that the pair (R', B') is obtained from (R, B) by a **symmetric exchange**. Using this terminology, we define the **exchange distance** (or **distance** for short) of two pairs of disjoint

[1]The work was supported by the Lendület Programme of the Hungarian Academy of Sciences – grant number LP2021-1/2021 and by the Hungarian National Research, Development and Innovation Office – NKFIH, grant numbers FK128673 and TKP2020-NKA-06.

[2]Tamás Schwarcz was supported by the ÚNKP-22-3 New National Excellence Program of the Ministry for Culture and Innovation from the source of the National Research, Development and Innovation Fund.

bases $\mathbf{P}_1 = (R_1, B_1)$ and $\mathbf{P}_2 = (R_2, B_2)$ to be the minimum number of symmetric exchanges needed to transform the former into the latter if such a sequence exists and $+\infty$ otherwise. We call two pairs of disjoint bases **equivalent** if their exchange distance is finite. A sequence of symmetric exchanges starting from a pair $\mathbf{P}_1$ is called **strictly monotone with respect to another pair** $\mathbf{P}_2$ (or **strictly monotone** for short when $\mathbf{P}_2$ is clear from the context) if each step decreases the difference between the first member of the current pair and that of $\mathbf{P}_2$. In other words, a strictly monotone exchange sequence uses elements only from $(R_1 \cap B_2) \cup (R_2 \cap B_1)$ and at most once.

At this point it is not clear (I) when the distance of two pairs will be finite, and (II) if their distance is finite, whether we can give an upper bound on it. Regarding question (I), one can formulate an obvious necessary condition for the distance of $\mathbf{P}_1$ and $\mathbf{P}_2$ to be finite, namely $R_1 \cup B_1 = R_2 \cup B_2$ should certainly hold – two pairs with this property are called **compatible**. In [19], White conjectured that *two basis pairs* $\mathbf{P}_1$ *and* $\mathbf{P}_2$ *are equivalent if and only if they are compatible.* While the conjecture was verified for various matroid classes, it remains open in general.

Much less is known about question (II), that is, the optimization version of the problem. Gabow [10] studied sequential symmetric exchanges and posed the following problem, which was later stated as a conjecture by Wiedemann [20] and by Cordovil and Moreira [7]: *for any two disjoint bases* R *and* B *of a matroid* M*, there is a sequence of* r *symmetric exchanges that transforms the pair* $\mathbf{P}_1 = (R, B)$ *into* $\mathbf{P}_2 = (B, R)$. The rank of the matroid is a trivial lower bound on the minimum number of exchanges needed to transform a pair (R, B) into (B, R), and the essence of Gabow's conjecture is that that many steps might always suffice. The relation between the conjectures of White and Gabow is immediate: the latter would imply the former for sequences of the form (R, B) and (B, R).

In general, if M has rank r, then $r - |R_1 \cap R_2|$ is an obvious lower bound on the exchange distance of $\mathbf{P}_1 = (R_1, B_1)$ and $\mathbf{P}_2 = (R_2, B_2)$. However, it might happen that more symmetric exchanges are needed even if M is a graphic matroid; see [8] for a counterexample. As a common generalization of the conjectures of White and Gabow, Hamidoune [7] proposed a rather optimistic variant stating that *the exchange distance of compatible basis pairs is at most the rank of the matroid.*

Let $w\colon S \to \mathbb{R}_+$ be a weight function on the elements of the ground set S. Given a pair (R, B) of disjoint bases, we define the **weight of a symmetric exchange** $R' \coloneqq R - e + f$ and $B' \coloneqq B + e - f$ to be $w(e) + w(f)$, that is, the sum of the weights of the exchanged elements. Analogously to the unweighted setting, we define the **weighted exchange distance** (or **weighted distance** for short) of two pairs of disjoint bases $\mathbf{P}_1 = (R_1, B_1)$ and $\mathbf{P}_2 = (R_2, B_2)$ to be the minimum total weight of symmetric exchanges needed to transform the former into the latter if such a sequence exists and $+\infty$ otherwise. As a weighted extension of Hamidoune's conjecture, we propose the following.

Conjecture 1 *Let* $\mathbf{P}_1 = (R_1, B_1)$ *and* $\mathbf{P}_2 = (R_2, B_2)$ *be compatible basis pairs of a matroid* M *over a ground set* S*, and let* $w\colon S \to \mathbb{R}_+$ *be a weight function. Then the weighted exchange distance of* $\mathbf{P}_1$ *and* $\mathbf{P}_2$ *is at most* $w(R_1 \cup B_1) = w(R_2 \cup B_2)$.

By setting the weight function to be identically one, we get back Hamidoune's conjecture. It is worth mentioning that a strictly monotone exchange sequence transforming $\mathbf{P}_1$ into $\mathbf{P}_2$ is optimal in every sense, i.e., it has both minimum length and minimum weight.

Previous work. By relying on the constructive characterization of bispanning graphs, Farber, Richter, and Shank [9] proved White's conjecture for graphic and cographic matroids, while Farber [8] settled the statement for transversal matroids. Bonin [6] verified the conjecture for sparse paving matroids. The case of strongly base orderable matroids was solved by Lasoń and Michałek [15]. McGuinness [17] extended the graphic case to frame matroids satisfying a certain linearity condition. Kotlar and Ziv [14] showed that any two elements of a basis have a sequential symmetric exchange with some two elements of any other basis. At the same time, Kotlar [13] proved that three consecutive symmetric exchanges exist for any two bases of a matroid, and that a full sequential symmetric exchange, of length at most 6, exists for any two bases of a matroid of rank 5.

Gabow's conjecture was verified for partition matroids, matching and transversal matroids, and matroids of rank less than 4 in [10], and an easy reasoning shows that it also holds for strongly base orderable

matroids as well. The graphic case was settled by Wiedemann [20], Kajitani, Ueno, and Miyano [12], and Cordovil and Moreira [7].

Recently, Bérczi and Schwarcz [3] showed that Hamidoune's conjecture holds for split matroids, a large class that contains paving matroids as well. While studying a specific maker-breaker game on bispanning graphs, Andres, Hochstättler and Merkel [1] showed that there is an exchange sequence between any two pairs of disjoint spanning trees of a wheel of rank at least four using only so-called left unique exchanges. They also asked whether the exchange distance of compatible basis pairs of a matroid can be bounded by a polynomial of the rank – this latter problem is a weakening of Hamidoune's conjecture.

The rank of the graphic matroid of a connected graph on n vertices is $n-1$. Though it is not stated explicitly, the algorithms of [9] and [5] that prove White's conjecture for graphic matroids give a sequence of exchanges of length at most $O(n^2)$. It remains an intriguing open problem to improve the bound to $O(n)$, matching the order of the bound in the conjecture.

Our results. We verify Conjecture 1 for various matroid classes. First we consider strongly base orderable matroids, a class with distinctive structural properties.

For the remaining matroid classes, we work with a further strengthening of the conjecture where both the length and the weight of the exchange sequence are ought to be bounded. We verify this stronger variant for split matroids, a class that was introduced only recently and generalizes paving matroids.

Our main result is a proof of the conjecture for graphic matroids of wheels. Though wheels are structurally rather simple, the proof for this graph class is already non-trivial and requires a thorough understanding of feasible exchanges. As a byproduct, we show that the minimum number of steps required to transform $\mathbf{P}_1$ into $\mathbf{P}_2$ can be arbitrarily large compared to the lower bound $r - |R_1 \cap R_2|$.

Finally, we prove the conjecture for spikes, an important class of 3-connected matroids. Spikes are interesting because, as we show, one can define an arbitrarily large number of basis pairs without a strictly monotone exchange sequence between any two of them. This is in sharp contrast to the case of wheels, where for any three pairs of bases, there exist two with a strictly monotone exchange sequence between them.

The rest of the paper is organized as follows. Basic notions and definitions are given in Section 2, together with some elementary observations on wheels. We verify Conjecture 1 for strongly base orderable matroids and for split matroids in Section 3. Graphic matroids of wheels are considered in Section 4, while spikes are discussed in Section 5.

Due to space constraints, several proofs and details are deferred to the full version of this paper, which is available at https://arxiv.org/abs/2211.12750.

2 Preliminaries

General notation. The set of nonnegative real numbers is denoted by $\mathbb{R}_+$. For subsets $X, Y \subseteq S$, their **symmetric difference** is defined as $X \triangle Y := (X \setminus Y) \cup (Y \setminus X)$. When Y consist of a single element y, then $X \setminus \{y\}$ and $X \cup \{y\}$ are abbreviated as $X - y$ and $X + y$, respectively. Given a weight function $w\colon S \to \mathbb{R}_+$ and a subset $X \subseteq S$, we use the notation $w(X) = \sum_{s \in X} w(s)$.

Matroids. For basic definitions on matroids, we refer the reader to [18]. If M is a rank-r matroid over a ground set S of size $2r$ such that S decomposes into two disjoint bases R and B of M, then such a decomposition is called a **coloring** of M[1]. A **feasible exchange** of elements $r \in R$ and $b \in B$ is denoted by (r, b). We extend this notation to a **sequence of symmetric exchanges** as well by writing $(r_1, b_1), \ldots, (r_k, b_k)$, meaning that $(R \setminus \{r_1, \ldots, r_i\}) \cup \{b_1, \ldots, b_i\}$ and $(B \cup \{r_1, \ldots, r_i\}) \setminus \{b_1, \ldots, b_i\}$ are bases for $i = 1, \ldots, k$. A matroid M is **strongly base orderable** if for any two bases B_1, B_2, there exists a bijection $\phi\colon B_1 \to B_2$ such that $(B_1 \setminus X) \cup \phi(X)$ is also a basis for any $X \subseteq B_1$, where we denote $\phi(X) := \{\phi(e) \mid e \in X\}$.

[1] Throughout the paper, we will refer to the elements of R and B as 'red' and 'blue', respectively.

Let S be a ground set of size at least r, $\mathcal{H} = \{H_1, \dots, H_q\}$ be a (possibly empty) collection of subsets of S, and $r, r_1, \dots, r_q$ be nonnegative integers satisfying $|H_i \cap H_j| \leq r_i + r_j - r$ for $1 \leq i < j \leq q$, and $|S \setminus H_i| + r_i \geq r$ for $i = 1, \dots, q$. Then $\mathcal{I} = \{ X \subseteq S \mid |X| \leq r,\ |X \cap H_i| \leq r_i \text{ for } 1 \leq i \leq q \}$ forms the family of independent sets of a rank-r matroid M that we call an **elementary split matroid**; see [2] for details. A set $F \subseteq S$ is called H_i**-tight** if $|F \cap H_i| = r_i$. A **split matroid** is the direct sum of a single elementary split matroid and some (maybe zero) uniform matroids.

For a graph $G = (V, E)$ on n vertices, the **graphic matroid** $M = (E, \mathcal{I})$ of G is defined on the edge set by considering a subset $F \subseteq E$ to be independent if it is a forest, that is, $\mathcal{I} = \{F \subseteq E \mid F \text{ does not contain a cycle}\}$. If the graph is connected, then the bases of the graphic matroid are exactly the spanning trees of G and the rank of the matroid is $n - 1$.

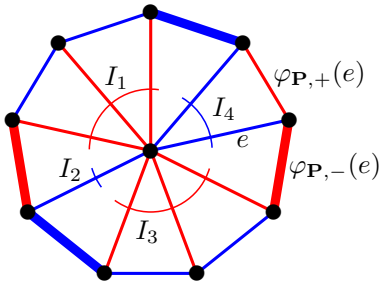
Let $S = \{t, x_1, y_1, \dots, x_r, y_r\}$ be a ground set of size $2r+1$ for some $r \geq 3$, and let $\mathcal{C}_1 = \{\{t, x_i, y_i\} \mid 1 \leq i \leq r\}$ and $\mathcal{C}_2 = \{\{x_i, y_i, x_j, y_j\} \mid 1 \leq i < j \leq r\}$. Furthermore, let $\mathcal{C}_3 \subseteq \{Z \subseteq S \mid |Z| = r,\ |Z \cap \{x_i, y_i\}| = 1 \text{ for } 1 \leq i \leq r\}$ be such that the intersection of any two members of $\mathcal{C}_3$ has size at most $r - 2$. Note that $\mathcal{C}_3$ might be empty. Finally, define $\mathcal{C}_4 = \{C \subseteq S \mid |C| = r + 1,\ C' \not\subseteq C \text{ for } C' \in \mathcal{C}_1 \cup \mathcal{C}_2 \cup \mathcal{C}_3\}$. Then the family $\mathcal{C} = \mathcal{C}_1 \cup \mathcal{C}_2 \cup \mathcal{C}_3 \cup \mathcal{C}_4$ satisfies the circuit axioms, hence $M = (S, \mathcal{C})$ is a rank-r matroid with circuit family $\mathcal{C}$. Matroids arising this way are called **spikes**, where t and the pairs $\{x_i, y_i\}$ are called the **tip** and the **legs** of the spike, respectively. It is not difficult to check that by restricting M to any $2r$ of its elements (or in other words, deleting any of its elements) results in a matroid whose ground set decomposes into two disjoint bases.

Wheels. A graph $G = (V, E)$ is called a **wheel graph** (or **wheel** for short) if it is obtained by connecting a vertex, called the **center of the wheel**, to all the vertices of a cycle of length at least three, called the **outer cycle** of the wheel. In particular, wheels have at least four vertices. Edges connecting the center vertex with the vertices of the outer cycle are called **spokes**, while the edges of the outer cycle are called **rim edges**. Wheels are clearly planar, and so the order of the vertices on the outer cycle implies a natural cyclic ordering of the spokes and the rim edges as well.

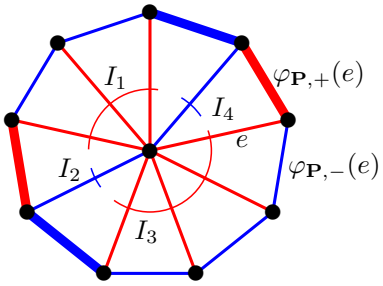
It is not difficult to check that any wheel is the disjoint union of two spanning trees. Therefore, a coloring of the graphic matroid of a wheel is basically a partition of its edge set into two colors R and B such that both color classes form a spanning tree. A nice property of wheels is that we have a fairly good understanding of the structure of their colorings. Indeed, in order to decompose a wheel into two spanning trees, we first need to split the spokes into two nonempty sets. Then, if a rim edge goes between the endpoints of two spokes having the same color, then it automatically has to be colored with the other color to obtain a basis. Hence it only remains to decide the color of the rim edges going between the endpoints of spokes having different colors. However, once we fix the color of any of those edges, it determines the color of all the remaining ones.

We call a maximal set of consecutive spokes of the same color an **interval**. By the **length** and **color of the interval** we mean the number and color of the spokes in it, respectively. Rim edges going between two intervals are called **boundary edges**. By the above, for $X \in \{R, B\}$, either every interval of color X is followed by a boundary edge of color X in a counterclockwise direction, or every interval of color X is followed by a boundary edge of color X in a clockwise direction. This property is referred to as the **orientation** of the coloring, and the orientation is called **positive** in the former and **negative** in the latter case; see Figure 1a for an example. Orientations will play a crucial role in whether one can go from one coloring to another using a small number of exchanges or not.

Given a coloring $\mathbf{P} = (R, B)$ of the wheel graph, each spoke e is incident to two rim edges. The rim edge sharing a vertex with e in the direction opposite of the orientation of the coloring can always be exchanged with e. We denote this rim edge by $\varphi_{\mathbf{P},-}(e)$, while the other rim edge incident to e is denoted by $\varphi_{\mathbf{P},+}(e)$; see Figure 1b for an example. Both $\varphi_{\mathbf{P},-}$ and $\varphi_{\mathbf{P},+}$ provide bijections between spokes and rim edges. Note that these bijections are already determined by the orientation of $\mathbf{P}$. If there are at least four intervals in the coloring, then the feasible exchanges are exactly the ones exchanging e and $\varphi_{\mathbf{P},-}(e)$ for some spoke e. When there are only two intervals in the coloring, then there are other pairs that can be symmetrically exchanged.



(a) The red and blue intervals are denoted by I_1, I_3 and I_2, I_4, respectively.



(b) The coloring obtained by symmetrically exchanging e and its pair $\varphi_{\mathbf{P},-}(e)$.

Figure 1: Colorings containing four intervals. Thick rim edges correspond to boundary edges. Note that both colorings have positive orientation.

3 Strongly base orderable and split matroids

As a warm-up, we first consider two basic cases: strongly base orderable and split matroids. For strongly base orderable matroids, the proof of Conjecture 1 can be read out directly from the proof of White's conjecture in [15] for strongly base orderable matroids. However, it might help the reader to get familiar with the notion of basis exchanges, and also sheds light to the difficulties caused by the presence of a weight function. For split matroids, the proof is more involved and relies heavily on that of Hamidoune's conjecture appeared in [3].

3.1 Strongly base orderable matroids

Theorem 2 *Let $\mathbf{P}_1 = (R_1, B_1)$ and $\mathbf{P}_2 = (R_2, B_2)$ be compatible pairs of disjoint bases of a rank-r strongly base orderable matroid M over a ground set S, and let $w\colon S \to \mathbb{R}_+$ be a weight function. Then there exists a sequence of exchanges of total weight at most $w(R_1 \cup B_1) = w(R_2 \cup B_2)$ that transforms $\mathbf{P}_1$ into $\mathbf{P}_2$ and uses each element at most twice.*

Proof: Let $\phi_1\colon R_1 \to B_1$ and $\phi_2\colon R_2 \to B_2$ be bijections such that $(R_i \setminus X) \cup \phi_i(X)$ is a basis for each $X \subseteq R_i$ and $i = 1, 2$. Consider the bipartite graph G with vertex set $R_1 \cup B_1 = R_2 \cup B_2$, and edges of the form $e\phi_1(e)$ for $e \in R_1$ and $f\phi_2(f)$ for $f \in R_2$. We denote the color classes of G by S and T. Note that

$$S = (S \cap R_1) \cup (S \cap B_1) = (R_1 \setminus (R_1 \setminus S)) \cup \phi_1(R_1 \setminus S)$$

and

$$T = (T \cap R_1) \cup (T \cap B_1) = (R_1 \setminus (R_1 \setminus T)) \cup \phi_1(R_1 \setminus T),$$

hence both S and T are bases of M. Let us define the basis pairs $\mathbf{P} = (S, T)$ and $\mathbf{P}' = (T, S)$.

By exchanging the elements between $R_1 \setminus S$ and $S \setminus R_1$ according to ϕ_1, we get a sequence of weight $w(R_1 \triangle S)$ that transforms $\mathbf{P}_1$ into $\mathbf{P}$. By exchanging the elements between $S \setminus R_2$ and $R_2 \setminus S$ according to ϕ_2, we get a sequence of weight $w(R_2 \triangle S)$ that transforms $\mathbf{P}$ into $\mathbf{P}_2$. The concatenation of these two sequences transforms $\mathbf{P}_1$ into $\mathbf{P}_2$, has total weight $w(R_1 \triangle S) + w(R_2 \triangle S)$, and uses each element at most twice.

By exchanging the elements between $R_1 \setminus T$ and $T \setminus R_1$ according to ϕ_1, we get a sequence of weight $w(R_1 \triangle T)$ that transforms $\mathbf{P}_1$ into $\mathbf{P}'$. By exchanging the elements between $T \setminus R_2$ and $R_2 \setminus T$ according to ϕ_2, we get a sequence of weight $w(R_2 \triangle T)$ that transforms $\mathbf{P}$ into $\mathbf{P}_2$. The concatenation of these two sequences transforms $\mathbf{P}_1$ into $\mathbf{P}_2$, has total weight $w(R_1 \triangle T) + w(R_2 \triangle T)$, and uses each element at most twice.

Since

$$\begin{aligned} & w(R_1 \triangle S) + w(R_2 \triangle S) + w(R_1 \triangle T) + w(R_2 \triangle T) \\ = & (w(R_1 \triangle S) + w(R_1 \triangle T)) + (w(R_2 \triangle S) + w(R_2 \triangle T)) \\ = & (w(R_1) + w(B_1)) + (w(R_2) + w(B_2)), \end{aligned}$$

at least one of the above defined sequences has total weight at most $w(R_1 \cup B_1) = w(R_2 \cup B_2)$ and uses each element at most twice. This concludes the proof of the theorem. □

3.2 Split matroids

The introduction of split matroids was motivated by the study of matroid polytopes from a geometry point of view [11]. Besides their immediate applications in tropical geometry, split matroids generalize paving matroids, a class that plays a fundamental role among matroids. In [3], Bérczi and Schwarcz showed that Hamidoune's conjecture holds for split matroids. In fact, they proved that the exchange distance of compatible basis pairs $\mathbf{P}_1 = (A_1, A_2)$ and $\mathbf{P}_2 = (B_1, B_2)$ of a rank-r split matroid is at most $\min\{r, r - |A_1 \cap B_1| + 1\}$, and that a shortest sequence transforming $\mathbf{P}_1$ into $\mathbf{P}_2$ can be found in polynomial time if the matroid in question is given by an independence oracle. By building on their proof, once can show how to deduce a strengthening of their result; see [4] for the details.

Theorem 3 *Let $\mathbf{P}_1 = (R_1, B_1)$ and $\mathbf{P}_2 = (R_2, B_2)$ be compatible pairs of disjoint bases of a rank-r split matroid M over a ground set S, and let $w\colon S \to \mathbb{R}_+$ be a weight function. Then there exists a sequence of exchanges of length at most $\min\{r, r - |R_1 \cap R_2| + 1\}$ and total weight at most $w(R_1 \cup B_1) = w(R_2 \cup B_2)$ that transforms $\mathbf{P}_1$ into $\mathbf{P}_2$ and uses each element at most twice.* □

4 Wheels

Our first main result is a proof of Conjecture 1 for the graphic matroid of wheels. In fact, we prove a much stronger statement: we verify that for any pair $\mathbf{P}_1 = (R_1, B_1)$ and $\mathbf{P}_2 = (R_2, B_2)$ of colorings of a wheel $G = (V, E)$, there exists a sequence of exchanges of length at most r and total weight at most $w(E)$ that transforms $\mathbf{P}_1$ into $\mathbf{P}_2$ and uses each edge at most twice.

Throughout the section, we assume that $\mathbf{P}_1$ has positive orientation. For ease of notation, we introduce $\varphi_\oplus \coloneqq \varphi_{\mathbf{P}_1,+}$ and $\varphi_\ominus \coloneqq \varphi_{\mathbf{P}_1,-}$. Recall that both $\varphi_\ominus$ and $\varphi_\oplus$ provide a bijection between spokes and rim edges.

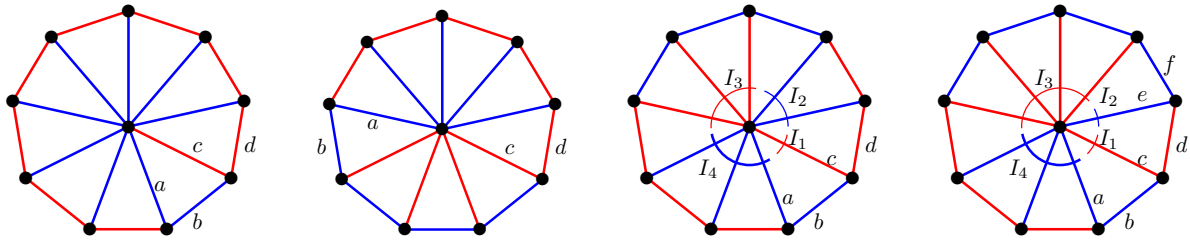
First we settle the case when the two colorings have the same orientation.

Lemma 4 *Let $\mathbf{P}_1 = (R_1, B_1)$ and $\mathbf{P}_2 = (R_2, B_2)$ be colorings of a wheel $G = (V, E)$ with the same orientation. Then there exists a strictly monotone sequence of exchanges that transforms $\mathbf{P}_1$ into $\mathbf{P}_2$.*

Proof: Exchange each spoke e that has different color in $\mathbf{P}_1$ than in $\mathbf{P}_2$ with its pair $\varphi_\ominus(e)$ in an arbitrary order, only paying attention to always have at least one spoke in both color classes. Once the spokes have the right colors, that is, they are colored as in $\mathbf{P}_2$, the rim edges are also colored as required. Indeed, the orientation was not changed during the procedure and the coloring of the spokes together with the orientation of the coloring uniquely determines the colors of the rim edges. □

Next we consider colorings with different orientations and a bounded number of intervals in one of the color classes.

Lemma 5 *Let $\mathbf{P}_1 = (R_1, B_1)$ and $\mathbf{P}_2 = (R_2, B_2)$ be colorings of a wheel $G = (V, E)$ with different orientations where $\mathbf{P}_1$ has at most four intervals, and let $w\colon E \to \mathbb{R}_+$ be a weight function. Then there exists a sequence of exchanges of length at most $n - 1$ and total weight at most $w(E)$ that transforms $\mathbf{P}_1$ into $\mathbf{P}_2$ and uses each edge at most twice.*



(a) $\mathbf{P}_1$ contains two intervals, one of them having length one.

(b) $\mathbf{P}_1$ contains two intervals, both having length at least two.

(c) $\mathbf{P}_1$ contains four intervals, one of them having length one.

(d) $\mathbf{P}_1$ contains both a red and a blue interval of length one.

Figure 2: Illustration of the cases in the proof of Lemma 5.

Proof: We distinguish two main cases and several subcases based on the number of intervals in $\mathbf{P}_1$. Due to space constraints, we only include the proof for some of the cases, see [4] for the missing ones. Recall that $\mathbf{P}_1$ is assumed to have positive orientation throughout.

Case 1. $\mathbf{P}_1$ has two intervals.

Case 1.1. $\mathbf{P}_1$ has an interval of length one.

We may assume that there exists a red interval of length one, and let c denote the unique spoke in it. We denote by a the spoke following c in negative direction, and further define $b \coloneqq \varphi_{\oplus}(a) = \varphi_{\ominus}(c)$ and $d \coloneqq \varphi_{\oplus}(c)$. Note that a and b are blue, while d is a red edge; see Figure 2a. As $\mathbf{P}_2$ has negative orientation, a and $\varphi_{\oplus}(a) = b$ have different colors in $\mathbf{P}_2$, and the same holds for c and $\varphi_{\oplus}(c) = d$. Hence the set of edges among a, b, c and d that have different colors in $\mathbf{P}_1$ and $\mathbf{P}_2$ is either $\{a, c\}, \{a, d\}, \{b, d\}$ or $\{b, c\}$. In the first three cases, changing the color of the two edges is a feasible exchange which reverses the orientation. Once the orientation of the coloring is reversed, there exists a strictly monotone exchange sequence to $\mathbf{P}_2$ by Lemma 4, altogether resulting in a strictly monotone exchange sequence from $\mathbf{P}_1$ to $\mathbf{P}_2$.

The only remaining case is when the set of edges among a, b, c and d that need to change color is $\{b, c\}$. In this case, the difficulty comes from the fact that these edges do not define a feasible exchange between the two color classes. In order to overcome this, extra steps are needed to reverse the orientation. Let s be an arbitrary red spoke of $\mathbf{P}_2$. Note that $s \notin \{a, c\}$ as we are in the case when a and c are blue in $\mathbf{P}_2$. As $\mathbf{P}_1$ has a unique red spoke, namely c, we get that s is blue in $\mathbf{P}_1$, and $\varphi_{\oplus}(s)$ is red in $\mathbf{P}_1$ and blue in $\mathbf{P}_2$ since $\mathbf{P}_2$ has negative orientation. Consider the two exchange sequences of length three (b, d), $(s, \varphi_{\oplus}(s))$, (c, d) and (a, c), $(s, \varphi_{\oplus}(s))$, (a, b). Both of these sequences reverse the orientation of the coloring and fix the colors of the edges a, b, c and d. Therefore, after applying any of them, there exists a strictly monotone exchange sequence to $\mathbf{P}_2$ by Lemma 4 that uses all the remaining edges in $E - \{a, b, c, d\}$ at most once. Thus in overall, we get an exchange sequence that uses each edge in $E - \{a, d\}$ at most once, does not use one of a and d and uses the other twice. Hence the length of the sequence is at most half of the number of edges, that is, $n - 1$. If $w(a) \geq w(d)$, then starting the sequence with (b, d), $(s, \varphi_{\oplus}(s))$, (c, d), while if $w(a) < w(d)$, then starting the sequence with (a, c), $(s, \varphi_{\oplus}(s))$, (a, b) ensures that total weight of the exchange sequence is at most $w(E)$, concluding the proof of the case.

Case 1.2. Both intervals of $\mathbf{P}_1$ have length at least two.

Let c denote the last spoke of the red interval in positive direction and let $d \coloneqq \varphi_{\oplus}(c)$. Furthermore, let a be the last spoke of the blue interval in positive direction and let $b \coloneqq \varphi_{\oplus}(a)$, see Figure 2b. Similarly to Case 1.1, the set of edges among a, b, c and d that have different colors in $\mathbf{P}_1$ and $\mathbf{P}_2$ is either $\{a, c\}$, $\{a, d\}$, $\{b, d\}$ or $\{b, c\}$. However, now fixing the orientation is even simpler than before as any of the exchanges (a, c), (a, d), (b, c) and (b, d) is feasible. After reversing the orientation using one of these exchanges, there exists a strictly monotone exchange sequence to $\mathbf{P}_2$ by Lemma 4, altogether resulting in a strictly monotone exchange sequence from $\mathbf{P}_1$ to $\mathbf{P}_2$.

We denote the number of spokes in R_1, R_2, B_1 and B_2 by r_1, r_2, b_1 and b_2, respectively. Let $2q$ denote the number of intervals in $\mathbf{P}_1$, and let $I_1, \dots, I_{2q}$ denote the intervals in a positive direction, where intervals with odd indices have color red and intervals with even indices have color blue. Furthermore, for $1 \le i \le 2q$, we define

$$x_i \coloneqq \sum[w(e) \mid e \in I_i \cup \varphi_\ominus(I_i),\ e \text{ has the same color in } \mathbf{P}_1 \text{ and } \mathbf{P}_2],$$
$$y_i \coloneqq \sum[w(e) \mid e \in I_i \cup \varphi_\ominus(I_i),\ e \text{ has different colors in } \mathbf{P}_1 \text{ and } \mathbf{P}_2].$$

By the above definitions, we have $w(I_i \cup \varphi_\ominus(I_i)) = x_i + y_i$ for $1 \le i \le 2q$, and $\sum_{i=1}^{2q}(x_i + y_i) = w(E)$.

The following cases can be solved in a similar manner.

Case 2. $\mathbf{P}_1$ has four intervals.

Since $r_1 + r_2 + b_1 + b_2 = 2(n-1)$, we have $\min\{r_1 + r_2, b_1 + b_2\} \le n - 1$. We may assume that $r_1 + r_2 \le n - 1$. We distinguish two cases based on the structure of the red intervals in $\mathbf{P}_1$.

Case 2.1. $\mathbf{P}_1$ has no red interval of length one.

Case 2.2. $\mathbf{P}_1$ has a red interval of length one. □

Our last technical lemma shows that when one of the colorings has at least six intervals, then there exists a sequence of exchanges that has low weight with respect to two arbitrary weight functions simultaneously.

Lemma 6 *Let $\mathbf{P}_1 = (R_1, B_1)$ and $\mathbf{P}_2 = (R_2, B_2)$ be colorings of a wheel $G = (V, E)$ with different orientations such that $\mathbf{P}_1$ has at least six intervals, and let $w_1, w_2 \colon E \to \mathbb{R}_+$ be weight functions. Then there exists a sequence of exchanges of total w_i-weight at most $w_i(E)$ for $i = 1, 2$ that transforms $\mathbf{P}_1$ into $\mathbf{P}_2$ and uses each edge at most twice.*

Proof: We distinguish two cases based on the remainder of the number of intervals modulo four.

Case 1. $q = 2k + 1$ for some integer $k \ge 1$.

For an index $1 \le j \le 4k + 2$, exchange each spoke $e \in \bigcup_{i=1}^{k} I_{j+2i-1}$ with its pair $\varphi_\ominus(e)$, and do the same for each spoke $e \in \bigcup_{i=k+1}^{2k} I_{j+2i}$. After these exchanges, the resulting coloring $\mathbf{P}_1'$ has two intervals: $I_j \cup I_{j+1} \cup \dots \cup I_{j+2k}$ has the same color in $\mathbf{P}_1'$ as I_j in $\mathbf{P}_1$, and $I_{j+2k+1} \cup I_{j+2k+2} \cup \dots \cup I_{j+4k+1}$ has the other color. Note that none of these two intervals has length one as $k \ge 1$. Therefore, there exists a strictly monotone exchange sequence from $\mathbf{P}_1'$ to $\mathbf{P}_2$ by Case 1.2 of Lemma 5. Let $w \in \{w_1, w_2\}$, and let us define I_i, x_i and y_i for $1 \le i \le 2q$ as in the proof of Lemma 5, where the x_i and y_i values are computed with respect to w. Our goal is to bound the w-weight of the above defined sequence of exchanges.

Exchanging each spoke e in $\bigcup_{i=1}^{k} I_{j+2i-1} \cup \bigcup_{i=k+1}^{2k} I_{j+2i}$ with its pair $\varphi_\ominus(e)$ has weight

$$\sum_{i=1}^{k}(x_{j+2i-1} + y_{j+2i-1}) + \sum_{i=k+1}^{2k}(x_{j+2i} + y_{j+2i}).$$

Then the strictly monotone sequence to $\mathbf{P}_2$ has weight

$$\sum_{i=0}^{k} y_{j+2i} + \sum_{i=1}^{k} x_{j+2i-1} + \sum_{i=k}^{2k} y_{j+2i+1} + \sum_{i=k+1}^{2k} x_{j+2i}.$$

The total weight is then

$$2 \cdot \left(\sum_{i=1}^{k} x_{j+2i-1} + \sum_{i=k+1}^{2k} x_{j+2i} \right) + \sum_{i=1}^{4k+2} y_i.$$

Therefore the total w-weight of the exchange sequence is at most $w(E) = \sum_{i=1}^{4k+2}(x_i + y_i)$ if and only if

$$\sum_{i=1}^{k} x_{j+2i-1} + \sum_{i=k+1}^{2k} x_{j+2i} \le \sum_{i=0}^{k} x_{j+2i} + \sum_{i=k}^{2k} x_{j+2i+1}. \qquad (\mathrm{A}_w(j))$$

Consider inequalities $\mathrm{A}_w(j)$ and $\mathrm{A}_w(j+1)$. The sum of these two inequalities gives

$$\left(\sum_{i=1}^{4k+2} x_i\right) - (x_j + x_{j+2k+1}) \leq \left(\sum_{i=1}^{4k+2} x_i\right) + (x_j + x_{j+2k+1}).$$

As this inequality clearly holds, at least one of $\mathrm{A}_w(j)$ and $\mathrm{A}_w(j+1)$ must hold as well. Furthermore, $\mathrm{A}_w(j)$ is identical to $\mathrm{A}_w(j+2k+1)$. These together imply that $\mathrm{A}_w(j)$ holds for at least $k+1$ choices of j from $\{1,\ldots,2k+1\}$ for $w \in \{w_1, w_2\}$. Therefore, there exists an index j for which both $\mathrm{A}_{w_1}(j)$ and $\mathrm{A}_{w_2}(j)$ are satisfied. As each edge is used at most twice, the statement follows.

Case 2. $q = 2k$ for some integer $k \geq 2$.

Our approach is similar to that of Case 1. For an index $1 \leq j \leq 4k$, exchange each spoke $e \in \bigcup_{i=1}^{k-1} I_{j+2i-1}$ with its pair $\varphi_\ominus(e)$, and do the same for each spoke $e \in \bigcup_{i=k}^{2k-1} I_{j+2i}$. After these exchanges, the resulting coloring $\mathbf{P}_1'$ has two intervals: $I_j \cup I_{j+1} \cup \cdots \cup I_{j+2k-1}$ has the same color in $\mathbf{P}_1'$ as I_j in $\mathbf{P}_1$, and $I_{j+2k} \cup I_{j+2k+1} \cup \cdots \cup I_{j+4k-1}$ has the other color. Note that none of these two intervals has length one as $k \geq 2$. Therefore, there exists a strictly monotone exchange sequence from $\mathbf{P}_1'$ to $\mathbf{P}_2$ by Case 1.2 of Lemma 5. Let $w \in \{w_1, w_2\}$, and let us define I_i, x_i and y_i for $1 \leq i \leq 2q$ as in the proof of Lemma 5, where the x_i and y_i values are computed with respect to w. Our goal is to bound the w-weight of the above defined sequence of exchanges.

Exchanging each spoke e in $\bigcup_{i=1}^{k-1} I_{j+2i-1} \cup \bigcup_{i=k}^{2k-1} I_{j+2i}$ with its pair $\varphi_\ominus(e)$ has weight

$$\sum_{i=1}^{k-1}(x_{j+2i-1} + y_{j+2i-1}) + \sum_{i=k}^{2k-1}(x_{j+2i} + y_{j+2i}).$$

Then the strictly monotone sequence to $\mathbf{P}_2$ has weight

$$\sum_{i=0}^{k-1} y_{j+2i} + \sum_{i=1}^{k-1} x_{j+2i-1} + \sum_{i=k}^{2k} y_{j+2i-1} + \sum_{i=k}^{2k-1} x_{j+2i}.$$

The total weight is then

$$2 \cdot \left(\sum_{i=1}^{k-1} x_{j+2i-1} + \sum_{i=k}^{2k-1} x_{j+2i}\right) + \sum_{i=1}^{4k} y_i.$$

Therefore the total w-weight of the exchange sequence is at most $w(E) = \sum_{i=1}^{4k}(x_i + y_i)$ if and only if

$$\sum_{i=1}^{k-1} x_{j+2i-1} + \sum_{i=k}^{2k-1} x_{j+2i} \leq \sum_{i=0}^{k-1} x_{j+2i} + \sum_{i=k-1}^{2k-1} x_{j+2i+1}. \qquad (\mathrm{B}_w(j))$$

Consider inequalities $\mathrm{B}_w(j)$ and $\mathrm{B}_w(j+1)$. The sum of these two inequalities gives

$$\left(\sum_{i=1}^{4k} x_i\right) - (x_j + x_{j+2k-1}) \leq \left(\sum_{i=1}^{4k} x_i\right) + (x_j + x_{j+2k-1}).$$

As this inequality clearly holds, at least one of $\mathrm{B}_w(j)$ and $\mathrm{B}_w(j+1)$ must hold as well. This implies that $\mathrm{B}_w(j)$ holds for at least $2k$ choices of j from $\{1,\ldots,4k\}$. Note that if the number of such choices is exactly $2k$, then $\mathrm{B}_w(j)$ holds either for all odd or for all even indices.

Now consider inequalities $\mathrm{B}_w(j)$ and $\mathrm{B}_w(j+2k)$. The sum of these two inequalities gives

$$\left(\sum_{i=1}^{4k} x_i\right) - (x_{j+2k-1} + x_{j+4k-1}) \leq \left(\sum_{i=1}^{4k} x_i\right) + (x_{j+2k-1} + x_{j+4k-1}).$$

As this inequality clearly holds, at least one of $\mathrm{B}_w(j)$ and $\mathrm{B}_w(j+2k)$ must hold as well. As the parities of j and $j+2k$ are the same, this, together with the above observation, implies that $\mathrm{B}_w(j)$ holds for at

least $2k+1$ choices of j from $\{1,\dots,4k\}$ for $w \in \{w_1, w_2\}$. Therefore, there exists an index j for which both $\mathrm{B}_{w_1}(j)$ and $\mathrm{B}_{w_2}(j)$ are satisfied. As each edge is used at most twice, the statement follows. □

With the help of Lemmas 4, 5 and 6, we are ready to prove the main result of the paper.

Theorem 7 *Let $\mathbf{P}_1 = (R_1, B_1)$ and $\mathbf{P}_2 = (R_2, B_2)$ be colorings of a wheel $G = (V, E)$, and let $w\colon E \to \mathbb{R}_+$ be a weight function. Then there exists a sequence of exchanges of length at most $n-1$ and total weight at most $w(E)$ that transforms $\mathbf{P}_1$ into $\mathbf{P}_2$ and uses each edge at most twice.*

Proof: If the colorings have identical orientation, then the theorem follows by Lemma 4. Indeed, the length and the weight of any strictly monotone sequence of exchanges that transforms $\mathbf{P}_1$ into $\mathbf{P}_2$ achieves the natural lower bounds $(n-1) - |R_1 \cap R_2| \le n-1$ and $w(R_1 \triangle R_2) \le w(E)$, respectively, and uses each edge at most once.

Hence assume that the colorings have different orientations. If $\mathbf{P}_1$ has at most four intervals, then the theorem immediately follows by Lemma 5. Otherwise, Lemma 6 with the choice $w_1 := w$ and $w_2 \equiv 1$ ensures the existence of a sequence of exchanges of total weight at most $w_1(E) = w(E)$ and length at most $w_2(E)/2 = |E|/2 = n-1$ that uses each edge at most twice, concluding the proof. □

5 Spikes

A strengthening of Conjecture 1 analogous to Theorem 7 holds for spikes as well. Consider a rank-r spike M over ground set S, and let $w\colon S \to \mathbb{R}_+$ be a weight function. We show that for any two compatible basis pairs $\mathbf{P}_1 = (R_1, B_1)$ and $\mathbf{P}_2 = (R_2, B_2)$, there exists a sequence of exchanges of length at most r and total weight at most $w(S)$ that transforms $\mathbf{P}_1$ into $\mathbf{P}_2$ and uses each element at most twice.

Recall that $S = \{t, x_1, y_1, \dots, x_r, y_r\}$, where t is the tip and $\{x_i, y_i\}$ for $1 \le i \le r$ are the legs of the spike. Hence $R_1 \cup B_1 = R_2 \cup B_2$ does not contain exactly one element s of S; for short, we say that the pairs $\mathbf{P}_1$ and $\mathbf{P}_2$ **miss** the element s. We distinguish two cases depending on whether this element is the tip of M or not.

Lemma 8 *Let $\mathbf{P}_1 = (R_1, B_1)$ and $\mathbf{P}_2 = (R_2, B_2)$ be compatible pairs of disjoint bases of a rank-r spike M over a ground set S missing the tip t, and let $w\colon S \to \mathbb{R}_+$ be a weight function. Then there exists a sequence of exchanges of length at most r and total weight at most $w(S-t)$ that transforms $\mathbf{P}_1$ into $\mathbf{P}_2$ and uses each element at most twice.* □

Lemma 9 *Let $\mathbf{P}_1 = (R_1, B_1)$ and $\mathbf{P}_2 = (R_2, B_2)$ be compatible pairs of disjoint bases of a rank-r spike M over a ground set S missing the non-tip element x_1, and let $w\colon S \to \mathbb{R}_+$ be a weight function. Then there exists a sequence of exchanges of length at most r and total weight at most $w(S-x_1)$ that transforms $\mathbf{P}_1$ into $\mathbf{P}_2$ and uses each element at most twice.* □

The two lemmas together implies the following theorem.

Theorem 10 *Let $\mathbf{P}_1 = (R_1, B_1)$ and $\mathbf{P}_2 = (R_2, B_2)$ be compatible pairs of disjoint bases of a rank-r spike M over a ground set S, and let $w\colon S \to \mathbb{R}_+$ be a weight function. Then there exists a sequence of exchanges of length at most r and total weight at most $w(S)$ that transforms $\mathbf{P}_1$ into $\mathbf{P}_2$ and uses each element at most twice.*

Proof: The theorem follows by combining Lemmas 8 and 9. □

References

[1] S. D. Andres, W. Hochstättler, and M. Merkel. On a base exchange game on bispanning graphs. *Discrete Applied Mathematics*, 165:25–36, 2014.

[2] K. Bérczi, T. Király, T. Schwarcz, Y. Yamaguchi, and Y. Yokoi. Hypergraph characterization of split matroids. *Journal of Combinatorial Theory, Series A*, 194:105697, 2023.

[3] K. Bérczi and T. Schwarcz. Exchange distance of basis pairs in split matroids. *arXiv preprint arXiv:2203.01779*, 2022.

[4] K. Bérczi, B. Mátravölgyi, and T. Schwarcz. Weighted exchange distance of basis pairs. *arXiv preprint arXiv:2211.12750*, 2022.

[5] J. Blasiak. The toric ideal of a graphic matroid is generated by quadrics. *Combinatorica*, 28(3):283–297, 2008.

[6] J. E. Bonin. Basis-exchange properties of sparse paving matroids. *Advances in Applied Mathematics*, 50(1):6–15, 2013.

[7] R. Cordovil and M. L. Moreira. Bases-cobases graphs and polytopes of matroids. *Combinatorica*, 13(2):157–165, 1993.

[8] M. Farber. Basis pair graphs of transversal matroids are connected. *Discrete Mathematics*, 73(3):245–248, 1989.

[9] M. Farber, B. Richter, and H. Shank. Edge-disjoint spanning trees: A connectedness theorem. *Journal of Graph Theory*, 9(3):319–324, 1985.

[10] H. Gabow. Decomposing symmetric exchanges in matroid bases. *Mathematical Programming*, 10(1):271–276, 1976.

[11] M. Joswig and B. Schröter. Matroids from hypersimplex splits. *Journal of Combinatorial Theory, Series A*, 151:254–284, 2017.

[12] Y. Kajitani, S. Ueno, and H. Miyano. Ordering of the elements of a matroid such that its consecutive w elements are independent. *Discrete Mathematics*, 72(1-3):187–194, 1988.

[13] D. Kotlar. On circuits and serial symmetric basis-exchange in matroids. *SIAM Journal on Discrete Mathematics*, 27(3):1274–1286, 2013.

[14] D. Kotlar and R. Ziv. On serial symmetric exchanges of matroid bases. *Journal of Graph Theory*, 73(3):296–304, 2013.

[15] M. Lasoń and M. Michałek. On the toric ideal of a matroid. *Advances in Mathematics*, 259:1–12, 2014.

[16] D. Mayhew, M. Newman, D. Welsh, and G. Whittle. On the asymptotic proportion of connected matroids. *European Journal of Combinatorics*, 32(6):882–890, 2011.

[17] S. McGuinness. Frame matroids, toric ideals, and a conjecture of White. *Advances in Applied Mathematics*, 118:102042, 2020.

[18] J. Oxley. *Matroid Theory*, volume 21 of *Oxford Graduate Texts in Mathematics*. Oxford University Press, Oxford, second edition, 2011.

[19] N. L. White. A unique exchange property for bases. *Linear Algebra and its Applications*, 31:81–91, 1980.

[20] D. Wiedemann. Cyclic base orders of matroids. Manuscript, 1984.

Pebble Game algorithms and their implementations

Péter Madarasi

Department of Operations Research, ELTE Eötvös Loránd University, and the ELKH-ELTE Egerváry Research Group on Combinatorial Optimization, Eötvös Loránd Research Network (ELKH), Pázmány Péter sétány 1/C, 1117 Budapest, Hungary.
madarasip@staff.elte.hu

Lóránt Matúz

Department of Operations Research, ELTE Eötvös Loránd University, Pázmány Péter sétány 1/C, 1117 Budapest, Hungary.
matuzl20@student.elte.hu

Abstract: A multigraph $G = (V, E)$ is (k, ℓ)-sparse if every subset $X \subseteq V$ of the vertices induces at most $\max\{k|X| - \ell, 0\}$ edges. Finding a largest (k, ℓ)-sparse subgraph is a well-studied, polynomial-time solvable problem, which is widely used in rigidity applications and serves as the basis of several combinatorial algorithms. We present a new implementation and compare it with the library called KINARI-web on a wide range of random and real-world datasets. The computational study shows that the new implementation is consistently faster by one order of magnitude. Furthermore, we propose several heuristics to fine-tune the free parameters of the algorithm and investigate their practical efficiency. We also implement an algorithm for finding k arc-disjoint r-arborescences in a digraph and k edge-disjoint spanning trees in an undirected graph, which corresponds to the case $\ell = k$. Finally, we give an improved algorithm for the case $\ell = 2k$ when the sparsity condition is required only for the subsets of vertices of size at least 3, which is a crucial necessary condition of $3D$ rigidity for $k = 3$. Our implementation is available at `https://lemon.cs.elte.hu/repos/sparseGraphs`, and it is proposed to be part of the LEMON library.

Keywords: (k, ℓ)**-sparse graphs, Pebble Game algorithms, LEMON library**

1 Introduction

An undirected multigraph $G = (V, E)$ is called (k, ℓ)-sparse if every subset of the vertices induces at most $\max\{0, k|X| - \ell\}$ edges. Essentially, this means that there are only a limited number of edges induced in each subset of the vertices, in other words, the graph is "uniformly sparse". Testing sparsity and finding a largest sparse subgraph are widely used tools in rigidity applications. The concept of sparsity also often occurs in combinatorial optimization, for example, a graph is (k, k)-sparse if and only if its edge set can be partitioned into k forests. Therefore, efficient algorithms for testing sparsity and their implementations are crucial.

Historical overview The definition of (k, ℓ)-sparse graphs was introduced in 1979 by Loréa [19] as an example of matroidal families. They have been studied intensively in the last decades, and it became apparent that they have a variety of applications. For example, (k, k)-tight graphs (that is, the largest (k, k)-sparse graphs) appeared in the classical results of Nash-Williams [21] and Tutte [25] as the characterization of the graphs that can be decomposed into k edge-disjoint spanning trees. Later, Laman [15] showed that $(2, 3)$-tight graphs are the generic minimally rigid graphs for bar-and-joint frameworks in the plane, and $(2, 3)$-spanning graphs are the rigid ones. Note that the complexity of testing rigidity in $3D$ is wide open. For a more detailed treatment of this area, the reader is referred to [8, 24].

[1]This research has been implemented with the support provided by the Ministry of Innovation and Technology of Hungary from the National Research, Development and Innovation Fund, financed under the ELTE TKP 2021-NKTA-62 funding scheme. The research was supported by the Ministry of Innovation and Technology NRDI Office within the framework of the Artificial Intelligence National Laboratory Program.

Previous work The family of the celebrated Pebble Game algorithms can find a maximum-weight (k,ℓ)-sparse subgraph in $O(nm)$ time and a largest one in $O(n^2)$ time [7, 16, 17, 1, 2]. These algorithms play a crucial role in rigidity applications and serve as the basis of several combinatorial algorithms. For recognizing planar $(2,3)$-tight graphs, called Laman graphs, there is an $O(n\log^3 n)$ algorithm due to Rollin, Schlipf and Schulz [22]. However, no faster algorithms than the Pebble Game are known for the entire range of parameters $k > 0$ and $0 \leq \ell < 2k$. To the best of our knowledge, the only (non-open-source) implementation aimed to recognize (k,ℓ)-sparse graphs is KINARI-web [11], due to Fox, Jagodzinski, Yang and Streinu. Another related implementation and algorithm are due to Cs. Király and A. Mihálykó [13, 14, 20, 12], which makes a (k,ℓ)-tight (hyper)graph (k,ℓ)-redundant.

Our results We present a new implementation of several versions of the Pebble Game algorithm and compare them with the library called KINARI-web [11] on a wide range of random and real-world molecular graphs. We show that our implementation of the Component Pebble Game algorithm is consistently faster by an order of magnitude. Furthermore, we propose several heuristics to fine-tune the free parameters of the Basic Pebble algorithm and investigate their practical efficiency. We also implement an algorithm for finding k arc-disjoint r-arborescences in a digraph and k edge-disjoint spanning trees in an undirected graph, which correspond to the case $\ell = k$. In addition, we implement an algorithm for covering the arc set of a digraph with k arc-disjoint branchings and for covering the edge set of a graph with k edge-disjoint forests. The algorithms use the Pebble Game algorithm to find a proper orientation of the graph, then we try to construct k arc-disjoint arborescences [23, p. 904-928]. Our implementation is available online [18], and it is proposed to be part of the LEMON library [3]. We also give an improved algorithm for the case $\ell = 2k$ when the sparsity condition is required only for the subsets of vertices of size at least 3. Note that the case $\ell = 2k$ is a crucial tool for testing the $3D$ rigidity of the so-called block and hole graphs with a single hole [9], and gives a necessary condition of $3D$ rigidity for $k = 3$.

The next section summarizes the basic definitions related to (k,ℓ)-sparsity. In Section 3, we improve the best-known algorithm for the special case of $\ell = 2k$. Then, we give an overview of our new implementation, and compare it with another implementation, called KINARI. Finally, we introduce and benchmark some heuristics to improve the running time of the Basic Pebble Game algorithm.

2 Definitions and the Pebble Game algorithms

A multigraph $G = (V, E)$ is (k,ℓ)-*sparse* if any subset X of the vertices induces at most $\max\{0, k|X| - \ell\}$ edges. In the special case $\ell = 2k$, we require this only for the subsets X of the vertices of size at least 3. Furthermore, if G is (k,ℓ)-sparse and it has exactly $(k|V| - \ell)$ edges, then it is called (k,ℓ)-*tight*. We say that G is (k,ℓ)-*spanning* if it contains a (k,ℓ)-tight subgraph that spans the entire vertex set V. A (k,ℓ)-*component* is a largest induced proper subgraph $G' = (V', E')$ of a (k,ℓ)-sparse graph which induces exactly $(k|V'| - \ell)$ edges. In this paper, we focus on the following four problems. 1) *Decision*: decide if G is sparse, tight, spanning, or none; 2) *Extraction*: extract a largest sparse subgraph from G; and 3) *Components*: find all (k,ℓ)-components of G. We restrict ourselves to the case $k > 0$ and $0 \leq \ell \leq 2k$ [16]. As we will see, the case $\ell = 2k$ needs to be treated separately.

Roughly speaking, the Pebble Game algorithms [16] process the edges of the input graph one by one, and either accept or reject each of them. The edge acceptance condition is checked by reorienting an inner digraph constructed from the accepted edges. Initially, k pebbles are placed on each vertex of the directed graph, and throughout the algorithm, the number of pebbles plus the number of outgoing arcs remains k on each vertex — and hence the pebbles are moved along the arc whenever it is reversed. An edge uv is accepted if the total number of pebbles on its endpoints u and v is more than ℓ, where ℓ is the parameter of the sparsity. When accepting an edge, it is inserted into the digraph oriented away from an endpoint containing at least one pebble, and we remove one pebble from the vertex entered by the new arc. The rules of the algorithm ensure that the inner digraph has an orientation, which — using the Orientation lemma [6] — ensures that the set of accepted edges forms a largest (k,ℓ)-sparse subgraph at the end of the execution. For a more detailed description of this algorithm, the reader is referred to [16].

Note that the number of pebbles plus the number of outgoing arcs is always k on each vertex, therefore, one can easily present the algorithm without using the concept of pebbles [2].

For $0 \leq \ell < 2k$, the algorithm finds a largest (k, ℓ)-sparse subgraph regardless of the order in which the edges are processed, because the (k, ℓ)-sparse subsets of the edges form the independent sets of a matroid. For the same reason, processing the edges in non-increasing order by their weights, the algorithms extracts a maximum-weight (k, ℓ)-sparse subgraph.

Note, however, that for $\ell = 2k$, the (k, ℓ)-sparse edges sets do not form the independent sets of a matroid, hence the algorithm finds an inclusion-wise maximal (k, ℓ)-sparse subgraph only.

The Component Pebble game is an improved version of the algorithm above in the case $0 \leq \ell < 2k$. The main ingredient of this enhanced version is that the rejection of an edge can be performed by checking whether it is induced by a (k, ℓ)-component of the graph formed by the accepted edges. The (k, ℓ)-components can be represented in such a way that rejecting an edge takes constant time, whereas accepting an edge takes linear time in the number of the vertices, hence the running time of the algorithm is $O(n^2)$ [16, 17, 1, 2]. The algorithm requires that the edges are processed in a specific order, therefore it is not clear whether this idea extends to the weighted case. An attempt was given in [17], but the analysis of the proposed data structure is not correct: the "bounded property" does not hold in general.

3 Algorithms for the case $\ell = 2k$

In this section, we present an algorithm for extracting an inclusion-wise maximal $(k, 2k)$-sparse subgraph which runs in time $O(kn^2m)$, then we give an improved algorithm running in $O(k^2nm)$ steps. Note that if $\ell = 2k$, then only the empty graph would be $(k, 2k)$-sparse with respect to the original definition. Therefore, we only require that the subgraphs on at least three vertices are (k, ℓ)-sparse. First of all, we need the following lemma.

Lemma 1 *Let $G = (V, E)$ be a $(k, 2k)$-sparse simple graph and let $u, v \in V$ two of its vertices. Assume that there is no edge between u and v. Let D be an orientation of G in which the outdegrees are at most k, and the outdegrees of u and v are zero. Then, $G + uv$ is sparse if and only if there exists a path from each vertex in $V \setminus \{u, v\}$ to a vertex distinct from u and v with outdegree smaller than k.*

Proof: Since G is $(k, 2k)$-sparse, G can always be oriented in such a way that the outdegree of each vertex is at most k, and the outdegrees of u and v are zero by the Orientation lemma [6].

Now, we prove the statement of the lemma. First, assume that there exists a path from each vertex $V \setminus \{u, v\}$ to a vertex with outdegree smaller than k distinct from u and v. For a subset X of the vertices containing u, v and a third vertex w, take a path from w to a vertex with outdegree smaller than k distinct from u and v, and reverse it. Since the outdegree of every vertex remains at most k, the outdegrees of u and v are zero, and the outdegree of w is smaller than k, we get that

$$i(X) \leq \sum_{x \in X} \mathrm{out}(x) < (|X| - 2)k = k|X| - 2k,$$

which means that X is not tight. Therefore, $G + uv$ is (k, ℓ)-sparse.

Second, assume that for a vertex w, there exist no paths from w to any vertices with outdegree smaller than k distinct from u and v. Let R denote the set of vertices reachable from w in D. We prove that $X := R \cup \{u, v\}$ is a tight set, which prevents the insertion of edge uv. Observe that the outdegree of every vertex in R is exactly k, while the outdegrees of u and v are zero. Therefore,

$$i(X) = \sum_{x \in X} \mathrm{out}(x) = (|X| - 2)k = k|X| - 2k,$$

which was to be shown. □

Note that one obtains a straightforward algorithm from this proof, proposed by Cs. Király, which runs in $O(kn^2m)$ time.

Now, we present an improved version of this algorithm for extracting an inclusion-wise maximal $(k, 2k)$-sparse subgraph running in $O(k^2nm)$ time, also based on Lemma 1. To achieve this, instead of traversing the graph n times to process an edge, we execute merely one BFS from the vertices with outdegree smaller than k in the *reversed* of the digraph built by the Pebble Game algorithm and check whether all vertices are reached.

The detailed description of the algorithm is the following.

Algorithm 1: Pebble Game for $\ell = 2k$, improved version

Input: A simple graph $G = (V, E)$ on at least three vertices and an integer $k > 0$.
Output: An inclusion-wise maximal $(k, 2k)$-sparse subgraph in G.
Method: Construct a new digraph D on the vertex set V without any arcs. Then, process each edge uv in an arbitrary order as follows.
Reorient D such that the outdegrees of u and v are zero, which takes at most $2k$ path reversals. Run a BFS in the reverse of D from the vertices with outdegree smaller than k, distinct from u and v.

- If every vertex in $V \setminus \{u, v\}$ is reached, then accept edge uv, and insert it into D with arbitrary orientation.
- Otherwise, there exists a vertex w which was not reached, and hence edge uv cannot be inserted by Lemma 1.

After each edge is processed, output the set of accepted edges.

Complexity: One reorientation requires at most $2k$ path reversal in time $O(k^2n)$, and the further BFS calls take $O(kn)$ time for each edge. Hence, the algorithm takes $O(k^2nm)$ steps in total.

4 An efficient implementation of the Pebble Game algorithms

In this section, we give a detailed description of our implementations and an in-depth practical comparison with a previous implementation. Also, we present some of the ideas we used to speed up the Basic Pebble Game algorithm in practice, and finally, we discuss different heuristics to fine-tune the order of edges in the Basic Pebble Game, which we can choose freely in the unweighted case.

4.1 The implemented algorithms

We implemented the following algorithms.

1. Basic Pebble Game [16] for the basic range of $k > 0$ and $0 \leq \ell < 2k$, including some practical improvements discussed later in Section 4.2.
2. Algorithm 1 for extracting an inclusion-wise maximal $(k, 2k)$-sparse graphs.
3. Unweighted Component Pebble Game [16, 17], which stores the vertex sets of the (k, ℓ)-components to improve the efficiency instead of their edge sets like in [17]. Furthermore, it also contains our terminating condition, discussed in the following section.
4. Finding k arc-disjoint r-arborescences and covering with k arc-disjoint branchings, based on the algorithms described in [23, p. 904-928].
5. Finding k edge-disjoint spanning trees and covering with k edge-disjoint forests, based on the algorithms described in [23, p. 904-928].

All of our algorithms provide loads of flexible options, which are configurable through a user-friendly interface. All the implementations provide step-by-step execution control, furthermore, lots of query functions make sure that all relevant information produced by the algorithms can be accessed.

4.2 Details of the Basic Pebble Game

Now we give some interesting details about our implementation of the Basic Pebble Game algorithms.

Terminating condition The edge acceptance condition is to have more than ℓ pebbles on the endpoints. Therefore, if the total number of pebbles drops to ℓ, then each remaining edge is surely rejected. Terminating the algorithm at this point reduces the running time significantly on dense graphs because the largest sparse subgraph is often found after processing a small portion of the edges — this means that the total number of pebbles drops to ℓ quickly.

Breadth-first search Since the most time-consuming part of the Basic Pebble Game is the graph traversal algorithm, we made it as efficient as possible. We changed the complete implementation of BFS in the LEMON library in the following way. The initializing of the BFS consists of clearing the queue and marking the vertices as non-visited. Whilst the original BFS of LEMON iterates through all vertices and marks them, our version iterates only on the elements of the queue.

4.3 Benchmark environment

Now we describe the exact environment of the benchmarks that are discussed later. All compared algorithms solved the extraction problem, that is, they extract a largest sparse subgraph. The parameters k and ℓ are tested for each possible pair until a limit $K = 7$ on k, that is, each pair (k, ℓ) with $1 \leq k \leq K$ and $0 \leq \ell < 2k$. For a fixed number n of vertices, 5 graphs were generated, and the running times represent the average of the graphs on n vertices for all k and ℓ.

The types of multigraphs we tested were the following.

1. **Rigid**: Rigid graphs in the plane, that is, $(2, 3)$-spanning graphs generated by the following process, suggested by Cs. Király and implemented by A. Mihálykó. Let T denote the union of three trees on the same vertex set. Replace each vertex $v \in T$ with a complete graph on $d_T(v) + 1$ vertices such that all except one new vertex is incident with exactly one of the edges of v.
2. **Tight**: (k, k)-tight graphs, which are generated as taking the union of the edge sets of k random (labeled) spanning trees on the same vertex set.
3. **Molecular**: Molecular graphs from the Protein Data Bank (`https://www.rcsb.org`). Their rigidity in $3D$ can be determined using the molecular conjecture [10], which boils down to finding six disjoint spanning trees in the graph obtained by adding every edge five times.
4. **Random**: Erdős-Rényi random multigraphs, meaning that each edge is independently included with a given probability p. We take $p = 0.2$ and $p = 0.6$.

These test instances cover all possible outputs of the Pebble Game algorithms, that is, sparse, tight, spanning and none of them.

The benchmarks mentioned above were executed on a computer with 32 GB RAM and an AMD Ryzen 9 3950X CPU, using Linux operating system.

4.4 Comparison of the implementations

In this section, we compare our implementation with a previous one, called KINARI-web [11, 4, 5], which is a software project implementing data structures and algorithms for rigidity analysis, with a special focus on applications in mechanical structures, abstract sparse graphs, and molecules.

We compare the running times of the following algorithms.

1. **KINARI**: The Unweighted Component Pebble Game implemented in KINARI.
2. **Old**: A naive implementation of the Basic Pebble Game algorithm implemented in LEMON. This version does not use the terminating condition based on the total number of remaining pebbles. Moreover, unlike the current version, it does not contain the fast initialization of the BFS algorithm.

3. **Basic**: A refined version of the Basic Pebble Game algorithm, implemented in LEMON. This version includes all improvements described in Section 4.2.
4. **Comp**: The Unweighted Component Pebble Game implemented in LEMON. It uses the concept of (k, ℓ)-components to reject edges in constant time, described precisely in [16]. This implementation also includes the ideas described in Section 4.2.

4.4.1 Benchmarks

Now we present the running times of the algorithms for the graph types mentioned above. In each figure, the horizontal axis represents the number of vertices of the graphs and the vertical axis shows the running time of the particular algorithms in seconds. For each set of instances, there are two figures next to each other, except for the random graphs. The left compares all algorithms, while the right includes only the implementations in LEMON — which tend to be the fastest.

Rigid graphs

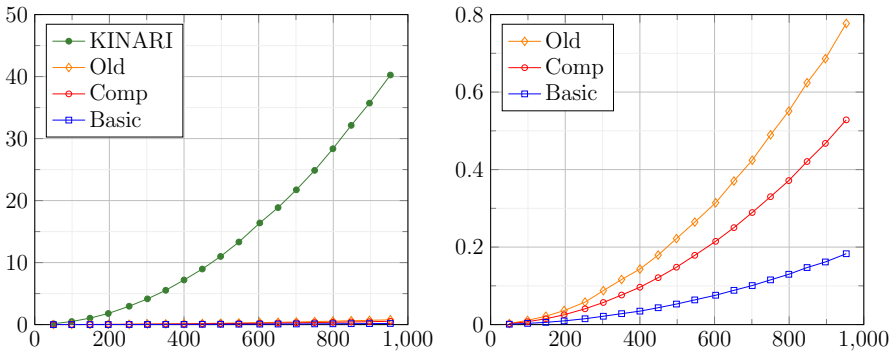


Figure 1: The comparison on rigid graphs. Based on the figure on the left, the three LEMON algorithms seem to be asymptotically faster than KINARI on rigid graphs. Moreover, the Component Pebble Game in LEMON seems to have larger running times than the Basic Pebble Game on rigid graphs. This is because it takes more time to update the (k, ℓ)-components than the extra graph traversals of the Basic Pebble Game.

Tight graphs

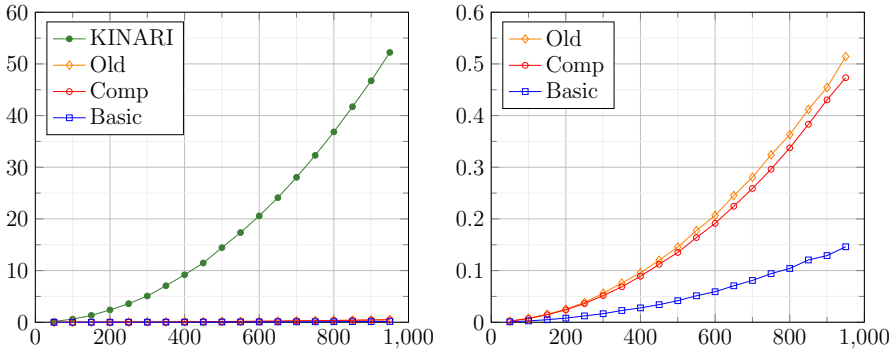


Figure 2: The comparison on tight graphs. The running times seem similar on tight and rigid graphs. However, the gap between the Component Pebble Game and the Basic Pebble Game is greater on tight graphs than on rigid ones. This is because no edge is rejected, which means that we update the data structures unnecessarily.

Molecular graphs

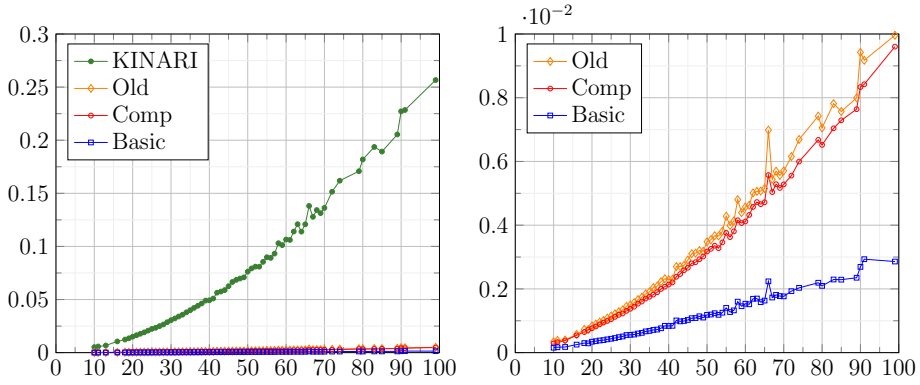
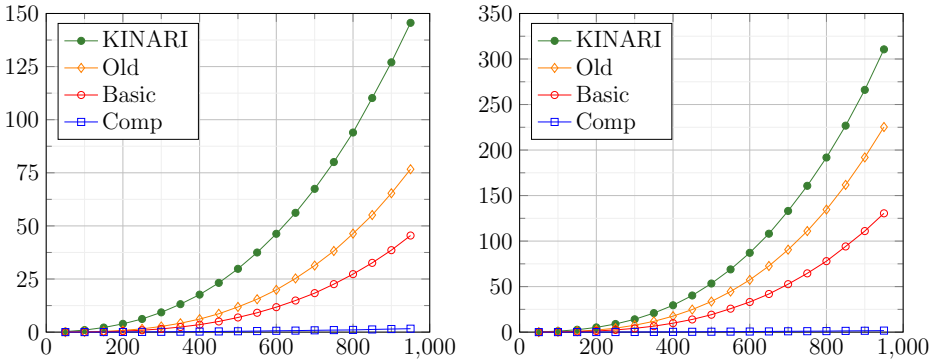


Figure 3: The comparison on molecular graphs. KINARI was principally developed for the rigidity analysis of molecular and protein graphs in $3D$. Nevertheless, it seems an order of magnitude slower than the LEMON versions on molecular graphs as well. Again, the Component Pebble Game algorithm is slower than the Basic Pebble Game on molecular graphs just like in the case of rigid and tight graphs.

Random graphs



(a) Random multigraphs with $p = 0.2$. (b) Random multigraphs with $p = 0.6$.

Figure 4: The comparison on random graphs with the given probabilities on the edges. Surprisingly, the running time of the Component Pebble Game implementation in KINARI is the closest to the Basic Pebble Game algorithm in LEMON — the former is supposed to be an order of magnitude faster than the latter on dense graphs. Furthermore, the denser the graphs are, the larger the gap grows between the Component Pebble Game in LEMON and the other algorithms. The reason behind this is that there are more and more edges to be rejected in the graphs, therefore, the constant-time rejections using the (k, ℓ)-components pay off. Moreover, the old version is also getting slower on dense random graphs, since the effect of our terminating condition increases heavily.

4.4.2 Summary of the benchmarks

The Component Pebble Game in LEMON is an order of magnitude faster than the Component Pebble Game of KINARI based on testing on a wide range of graphs. We also saw that the running time of the implementation in KINARI heavily depends on the density of the graphs, unlike the Component Pebble Game algorithm implemented in LEMON. In fact, the running time of the Component Pebble Game algorithm in KINARI seems much larger than the running times of any version implemented in LEMON.

The Component Pebble Game is outstanding for dense graphs and the Basic Pebble Game for sparse graphs, as we expected based on the worst-case analysis. We note that our experiment shows that the running time of Algorithm 1 for the case $\ell = 2k$ is similar to that of the Basic Pebble game for $\ell = 2k-1$.

4.5 Heuristic edge ordering

The running time of the Basic Pebble Game algorithm heavily depends on the processing order of the edges, because the algorithm terminates as soon as the largest sparse subgraph is found. In this section, we propose multiple heuristics for finding an order of the edges in which the algorithm processes the edges more efficiently. The basic idea is as follows. The edge acceptance condition for an edge uv depends on the total number of pebbles on its endpoints u and v. This means that the edges that have a large number of pebbles on their endpoints are more "likely" to be accepted by the Pebble Game algorithms. For example, if there is an edge that is insertable without any pebble collection, then we should choose that one. Therefore, we design heuristics that prioritize the edges with many pebbles on their endpoints, in the hope of finding a largest sparse subgraph and terminating as soon as possible.

4.5.1 The tested heuristics

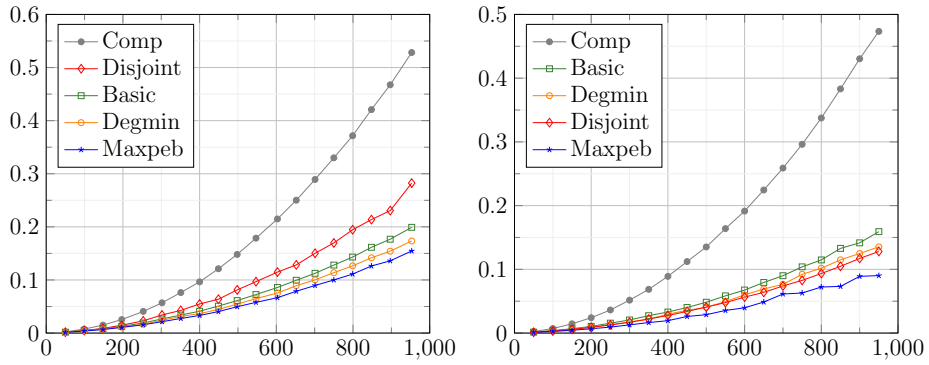
The tested heuristics were the following.

1. **Basic** and **Comp**: the most effective versions of the Basic and the Component Pebble Game implemented in LEMON, respectively. They process the edges in the order in which they appear in the graph representation, that is, they iterate over the vertices and process the incident edges with each vertex. This edge order is to be considered as the baseline in our experiments.
2. **Degmin**: select an edge incident with a vertex that has the smallest degree.
3. **Disjoint**: select an edge that has the fewest incident edges.
4. **Maxpeb**: select an edge such that the total number of pebbles on the endpoints of the edge is maximal.

4.5.2 Benchmarks

Now we present a practical study of the heuristics for the types of graphs mentioned above. Note that the presented running times do not include the time needed to select the next edge to be processed.

Rigid and tight graphs



(a) Rigid multigraphs.

(b) Tight multigraphs.

Figure 5: The comparison of the heuristics in rigid and tight graphs. The running times of the heuristic seem to be similar on both rigid and tight graphs, because a largest sparse subgraph is found almost at the same time regardless of the order of the edges as only a few of them are rejected.

Random graphs

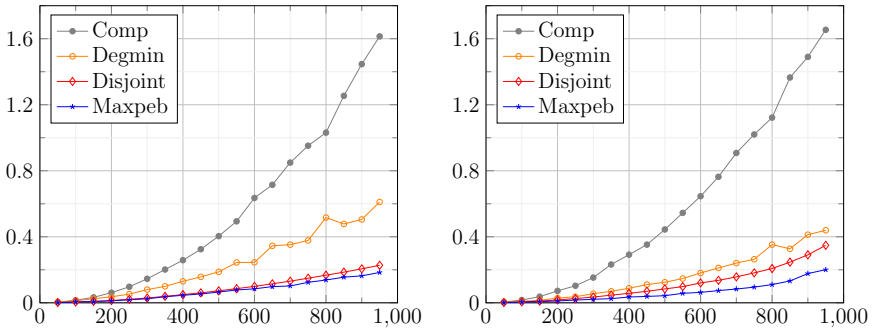


Figure 6: Random graphs with probabilities 0.2 and 0.6 on the edges, respectively. The Basic edge order is not shown, as it was utterly slower than any other algorithm for random graphs, see Figure 4. Moreover, observe that the running time of the Basic algorithm decreases, but that of the other heuristics increases on denser graphs. The figures on the right represent the differences between the rest of the heuristics. Selecting a vertex with a minimal degree seems to be the slowest among them. The second fastest heuristic is selecting a disjoint edge. Observe that the running time of this heuristic does not depend on the density of the graphs. The quickest heuristic on random graphs is selecting an edge with a maximum number of pebbles on the endpoints.

4.5.3 Conclusion of the heuristics

The order of the edges is proven highly decisive in speeding up the Pebble Game algorithms. The heuristic edge orders make the execution of the Basic Pebble Game algorithm one order of magnitude faster on dense graphs, similarly to the Component Pebble Game algorithm. Among the proposed heuristics, Maxpeb seems to be the most efficient, which supports our intuition.

Acknowledgement

The authors are grateful to Tibor Jordán, Csaba Király and András Mihálykó for the discussions and for pointing to relevant literature.

References

[1] A. R. Berg. *Rigidity of Frameworks and Connectivity of Graphs.* PhD thesis, Aarhus University, Denmark, 2004.

[2] A. R. Berg and T. Jordán. Algorithms for graph rigidity and scene analysis. In G. Di Battista and U. Zwick, editors, *Algorithms-ESA 2003: 11th Annual European Symposium, Budapest, Hungary, September 16-19, 2003. Proceedings 11*, pages 78–89. Springer LNCS 2832, 2003.

[3] B. Dezső, A. Jüttner, and P. Kovács. LEMON–An open source C++ graph template library. *Electronic notes in theoretical computer science*, 264(5):23–45, 2011.

[4] N. Fox, F. Jagodzinski, Y. Li, and I. Streinu. KINARI-Web: a server for protein rigidity analysis. *Nucleic acids research*, 39(suppl_2):W177–W183, 2011.

[5] N. Fox, F. Jagodzinski, and I. Streinu. KINARI-Lib: A C++ library for mechanical modeling and pebble game rigidity analysis. *Minisymposium on Publicly Available Geometric/Topological Software*, pages 29–32, 2012.

[6] S. L. Hakimi. On the degrees of the vertices of a directed graph. *Journal of the Franklin Institute*, 279(4):290–308, 1965.

[7] D. J. Jacobs and B. Hendrickson. An algorithm for two-dimensional rigidity percolation: the pebble game. *Journal of Computational Physics*, 137(2):346–365, 1997.

[8] T. Jordán. Combinatorial rigidity: graphs and matroids in the theory of rigid frameworks. *Discrete Geometric Analysis, MSJ Memoirs*, 34:33–112, 2016.

[9] T. Jordán. Rigid block and hole graphs with a single block. *Discrete Mathematics*, 346(3):113268, 2023.

[10] N. Katoh and S. Tanigawa. A proof of the molecular conjecture. *Discrete & Computational Geometry*, 45(4):647–700, 2011.

[11] KINARI. Kinematics and rigidity. `http://kinari.cs.umass.edu`. Accessed: April 27, 2022.

[12] Cs. Király and A. Mihálykó. Fast algorithms for sparsity matroids and the global rigidity augmentation problem. Technical Report TR-2022-05, Egerváry Research Group, Budapest, 2022. `https://egres.elte.hu`.

[13] Cs. Király and A. Mihálykó. Globally rigid augmentation of rigid graphs. *SIAM Journal on Discrete Mathematics*, 36(4):2473–2496, 2022.

[14] Cs. Király and A. Mihálykó. Sparse graphs and an augmentation problem. *Mathematical Programming*, 192(1-2):119–148, 2022.

[15] G. Laman. On graphs and rigidity of plane skeletal structures. *Journal of Engineering mathematics*, 4(4):331–340, 1970.

[16] A. Lee and I. Streinu. Pebble game algorithms and sparse graphs. *Discrete Mathematics*, 308(8):1425–1437, 2008.

[17] A. Lee, I. Streinu, and L. Theran. Finding and maintaining rigid components. *Canadian Conference on Computational Geometry*, 2005.

[18] LEMON. Library for Efficient Modeling and Optimization in Networks. Repository of sparse graphs: `https://lemon.cs.elte.hu/repos/sparseGraphs`. Accessed: February 21, 2023.

[19] M. Lorea. On matroidal families. *Discrete Mathematics*, 28(1):103–106, 1979.

[20] A. Mihálykó. `https://github.com/mihalykoandras/rigidityAugmentations.git`. Accessed: February 21, 2023.

[21] C. St. J. A. Nash-Williams. Edge-disjoint spanning trees of finite graphs. *Journal of the London Mathematical Society*, 1(1):445–450, 1961.

[22] J. Rollin, L. Schlipf, and A. Schulz. Recognizing Planar Laman Graphs. In M. A. Bender, O. Svensson, and G. Herman, editors, *27th Annual European Symposium on Algorithms (ESA 2019)*, volume 144 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 79:1–79:12, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[23] A. Schrijver. *Combinatorial optimization: polyhedra and efficiency*, volume B. Springer, 2003.

[24] B. Schulze and W. Whiteley. Rigidity and scene analysis. In *Handbook of Discrete and Computational Geometry*, pages 1593–1632. Chapman and Hall/CRC, 2017.

[25] W. T. Tutte. On the problem of decomposing a graph into n connected factors. *Journal of the London Mathematical Society*, 1(1):221–230, 1961.

Newton-type algorithms for inverse optimization problems I and II: Weighted infinity norm and span

Kristóf Bérczi[1]

MTA-ELTE Matroid Optimization Research Group
ELKH-ELTE Egerváry Research Group
Department of Operations Research
Eötvös Loránt University
Budapest, Hungary
kristof.berczi@ttk.elte.hu

Lydia Mirabel Mendoza-Cadena[1]

MTA-ELTE Matroid Optimization Research Group
Department of Operations Research
Eötvös Loránt University
Budapest, Hungary
lmmendoza@proton.me

Kitti Varga[12]

MTA-ELTE Matroid Optimization Research Group
ELKH-ELTE Egerváry Research Group
Department of Operations Research
Eötvös Loránt University
Budapest, Hungary
vkitti@math.bme.hu

Abstract: Inverse optimization problems appear naturally in diverse applications, such as system identification in seismic and medical tomography, or bilevel programming. In such problems, we are given a feasible but not necessarily optimal solution to an underlying optimization problem together with a linear cost function, and the goal is to modify the costs by a small deviation vector so that the input solution becomes optimal.

The difference between the new and the original cost functions can be measured in several ways. In this work, we focus on two variants: *Part I* concentrates on minimizing the weighted ℓ_∞-norm of the deviation vector, while *Part II* concentrates on minimizing its weighted span. In both cases, we provide a min-max characterization for the minimum size of an optimal deviation vector with respect to the given objective. Furthermore, we give a simple, purely combinatorial algorithm that determines such a vector in pseudo-polynomial time, assuming that a pseudo-polynomial time algorithm for solving the underlying combinatorial optimization problem is available.

Keywords: Algorithms, Infinity norm, Inverse optimization, Min-max theorem, Span

1 Introduction

Inverse optimization problems have long been the focus of research due to their wide applicability in both theory and practice. The roots of inverse optimization go back to the work of Burton and Toit [5] who

[1]The work was supported by the Lendület Programme of the Hungarian Academy of Sciences – grant number LP2021-1/2021 and by the Hungarian National Research, Development and Innovation Office – NKFIH, grant numbers FK128673 and TKP2020-NKA-06.

[2]Kitti Varga was supported by the Hungarian National Research, Development and Innovation Office – NKFIH, grant number K124171.

studied the inverse shortest paths problem, that is, the problem of recovering the edge costs given some information about the shortest paths in the graph. Since their pioneering work, countless of applications and extensions emerged; we refer the interested reader to [14] for the basics and to [7, 10] for surveys.

In a classical optimization problem, we are given a set of feasible solutions together with a linear cost function, and the goal is to find a feasible solution that minimizes or maximizes the cost. In contrast, in an inverse optimization problem we are also given a fixed feasible solution, and the goal is to modify the costs 'as little as possible' so that the input solution becomes optimal. There may be various ways to measure the deviation of the new cost function from the original one, and, as one would expect, the choice of the objective greatly affects the complexity of the problem. In order to avoid confusion, we refer to solutions of the inverse optimization problem and of the underlying combinatorial optimization problem as *feasible deviation vectors* and *solutions*, respectively.

In the past decades, inverse optimization problems found numerous applications. As an example, let us briefly describe the *Pathway concordance problem*, see [6]. A clinical pathway describes a standardized sequence of steps for managing a clinical process in the delivery of care for a specific disease, with the aim of optimizing the outcome on a patient or population-level. These processes are determined by multidisciplinary medical experts, and have been shown to efficiently improve e.g. patient survival and satisfaction, wait times, and cost of care. However, patients' journeys through the healthcare system can differ significantly from the recommended pathways, which raises the problem of measuring the concordance of patient-traversed pathways against the recommended ones. The problem can be modeled by a directed graph whose vertices correspond to activities that the patient can undertake, and the arcs indicate that a patient went from one activity to another. The 'cost' of a patient undertaking or missing certain activities and traversing arcs can be modeled by arc costs. The goal is to determine arc costs such that the reference pathways are optimal, that is, they are shortest paths between the corresponding start and end vertices. Then, assuming such arc costs are available, the journey of any patient can be scored based on the cost of the associated directed walk through the network.

Other applications include tomographic imaging [8], timely decision-making [11], and inverse transport that plays an important role e.g. in medical imaging (optical tomography, optical molecular imaging) and in geophysical imaging (remote sensing in the atmosphere), as explained in [3]. In all cases, the function used to measure the size of the deviation vector may be different depending on the actual problem. Two natural objectives are to minimize the largest absolute value of the coordinates of the deviation vector, and to minimize the difference between its largest and smallest coordinates. The former motivates the investigation of the ℓ_∞-norm objective, while the latter leads to the study of the span objective.

Previous work. Inverse problems under the ℓ_∞-norm have been studied in various settings. Xiaoguang [15] considered the inverse optimization problem of submodular functions on digraphs, and gave an LP-based algorithm that solves most inverse network optimization problems in polynomial time. Zhang and Liu [18] suggested a method for solving a general inverse LP problem including upper and lower bound constraints. Their approach is based on the optimality conditions for LP problems if the given feasible solution is a 0-1 vector, and one optimal solution of the original LP problem has all components between 0 and 1, which often happens in network or combinatorial optimization problems. In a later paper [13], the same authors studied the inverse maximum-weight matching problem in non-bipartite graphs under the ℓ_∞-norm objective. They showed that the problem can be formulated as a maximum-mean alternating cycle problem in an undirected network, and can be solved in polynomial time by a binary search algorithm and in strongly polynomial time by an ascending algorithm.

Using LP descriptions, Ahuja and Orlin [2] proved that if an optimization problem can be modeled as an LP, then the same holds for the underlying inverse optimization problem under ℓ_1- or ℓ_∞-norm objectives. Furthermore, if the optimization problem is polynomially solvable for linear cost functions, then the inverse counterparts with ℓ_1- and ℓ_∞-norms are also polynomially solvable.

In [19], Zhang and Liu proposed a model that generalizes numerous inverse combinatorial optimization problems when no bounds are given on the coordinates of the deviation vector. They exhibited a Newton-type algorithm for their model under the ℓ_∞-norm that solves the problem in strongly polynomial time, assuming that an underlying subproblem is solvable in strongly polynomial time for any fixed value of a

certain parameter that the subproblem depends on. In general, the problem arising is NP-hard, but it can be solved efficiently in special cases, such as the inverse spanning tree, shortest path, matching, or matroid intersection problems.

Zhang, Guan, and Zhang [17] provided a mathematical model of the inverse spanning tree problem, gave a characterization of optimal solutions, and developed a strongly polynomial-time algorithm for determining an optimal deviation vector. Yang and Zhang [16] presented strongly polynomial-time algorithms to solve the inverse min-max spanning tree and the inverse maximum capacity path problems when bounds are also given on the coordinates of the deviation vector. Lasserre [12] considered the inverse optimization problem associated with the polynomial program and a given current feasible solution, and provided a systematic numerical scheme to compute an inverse optimal solution. Ahmadian, Bhaskar, Sanità, and Swamy [1] studied integral inverse optimization problems from an approximation point of view. They obtained tight or nearly-tight approximation guarantees for various inverse optimization problems, and some of their results apply for ℓ_∞-norm as well. Recently, inverse optimization problems with multiple weight functions were introduced by the authors, see [4].

Most papers on inverse optimization consider algorithmic aspects, and so they do not provide a min-max characterization for the optimum value in question. Recently, Frank and Murota [9] developed a general min-max formula for the minimum of an integer-valued separable discrete convex function, where the minimum is taken over the set of integral elements of a box total dual integral polyhedron. Their approach covers and even extends a wide class of inverse combinatorial optimization problems. Nevertheless, our problems do not fit in the box-TDI framework as neither the ℓ_∞-norm nor the span is separable convex, and the optimal solutions are not necessarily integral for them.

Basic notation. We denote the sets of *real* and *positive real* numbers by $\mathbb{R}$ and $\mathbb{R}_+$, respectively. For a positive integer k, we use $[k] \coloneqq \{1, \dots, k\}$. Given a ground set S and subsets $X, Y \subseteq S$, the *symmetric difference* of X and Y is denoted by $X \triangle Y \coloneqq (X \setminus Y) \cup (Y \setminus X)$. For a weight function $w \in \mathbb{R}_+^S$, the total sum of its values over X is denoted by $w(X) \coloneqq \sum \{w(s) \mid s \in X\}$, where the sum over the empty set is always considered to be 0. Furthermore, we use $\frac{1}{w}(X) \coloneqq \sum \{\frac{1}{w(s)} \mid s \in X\}$. By convention, we define $\min\{\emptyset\} = +\infty$ and $\max\{\emptyset\} = -\infty$.

Our results. Let S be a finite ground set, $\mathcal{F} \subseteq 2^S$ be a collection of *feasible solutions* for an underlying optimization problem, $F^* \in \mathcal{F}$ be an *input solution*, $c \in \mathbb{R}^S$ be a *cost function*, $w \in \mathbb{R}_+^S$ be a positive *weight function*, and $\ell \colon S \to \mathbb{R} \cup \{-\infty\}$ and $u \colon S \to \mathbb{R} \cup \{+\infty\}$ be lower and upper bounds, respectively, such that $\ell \leq u$. In the *minimum-cost inverse optimization problem under weighted ℓ_∞-norm objective* $(S, \mathcal{F}, F^*, c, \ell, u, \|\cdot\|_{\infty,w})$, we seek a *deviation vector* $p \in \mathbb{R}^S$ such that

(a) F^* is a minimum cost member of $\mathcal{F}$ with respect to $c - p$,

(b) p is within the bounds $\ell \leq p \leq u$, and

(c) $\|p\|_{\infty,w} \coloneqq \max\{w(s) \cdot |p(s)| \mid s \in S\}$ is minimized.

In the *minimum-cost inverse optimization problem under weighted span objective* $\big(S, \mathcal{F}, F^*, c, \ell, u, \operatorname{span}_w(\cdot)\big)$, condition (c) modifies to

(c') $\operatorname{span}_w(p) \coloneqq \max\{w(s) \cdot p(s) \mid s \in S\} - \min\{w(s) \cdot p(s) \mid s \in S\}$ is minimized.

Due to the lower and upper bounds ℓ and u, it might happen that there exists no deviation vector p satisfying the requirements. A deviation vector is called *feasible* if it satisfies conditions (a) and (b), and *optimal* if in addition it attains the minimum in (c) or (c'). We denote the problems by $(S, \mathcal{F}, F^*, c, -\infty, +\infty, \|\cdot\|_{\infty,w})$ and $\big(S, \mathcal{F}, F^*, c, -\infty, +\infty, \operatorname{span}_w(\cdot)\big)$ when no bounds are given on the coordinates of p at all.

For the problems above, we provide min-max characterizations for the optimum value when $\ell \equiv -\infty$ and $u \equiv +\infty$. Our main result is giving purely combinatorial algorithms that determines an optimal deviation vector in pseudo-polynomial time, assuming access to a pseudo-polynomial time algorithm for the underlying combinatorial optimization problem. However, if $w \equiv 1$ and the underlying optimization

problem can be solved in (strongly) polynomial time, then our algorithms run in (strongly) polynomial time as well. Furthermore, the algorithms also work when the lower and upper bounds are arbitrary, and the feasible solutions of the underlying optimization problem have different sizes. Hence our framework includes classical inverse optimization problems such as the inverse spanning arborescence, matching, and matroid intersection problems under the ℓ_∞-norm and span objectives.

Although being rather similar in nature, the ℓ_∞-norm and the span behave quite differently as the infinite norm measures how far the coordinates of p are from 0, while the span measures how far the coordinates of p are from each other. In particular, it might happen that there exists a non-zero feasible deviation vector p with $\operatorname{span}(p) = 0$.

In the rest of the paper, results on the infinity norm objective are discussed in Section 2, while Section 3 considers the span objective. Due to space constraints, most of the proofs and details are deferred to the full version of this paper, which will soon be available on arXiv.

2 Weighted infinity norm

First, we consider the problem of minimizing the weighted ℓ_∞-norm of the deviation vector, where $w \in \mathbb{R}^S_+$ is a positive weight function. For any $\delta \geq 0$, let $p_{[\delta|\ell,u|w]} \colon S \to \mathbb{R}$ be defined as

$$p_{[\delta|\ell,u|w]}(s) \coloneqq \begin{cases} \ell(s) & \text{if } s \in F^* \text{ and } \delta/w(s) < \ell(s), \\ \delta/w(s) & \text{if } s \in F^* \text{ and } \ell(s) \leq \delta/w(s) \leq u(s), \\ u(s) & \text{if } s \in F^* \text{ and } u(s) < \delta/w(s), \\ \ell(s) & \text{if } s \in S \setminus F^* \text{ and } -\delta/w(s) < \ell(s), \\ -\delta/w(s) & \text{if } s \in S \setminus F^* \text{ and } \ell(s) \leq -\delta/w(s) \leq u(s), \\ u(s) & \text{if } s \in S \setminus F^* \text{ and } u(s) < -\delta/w(s). \end{cases}$$

We simply write $p_{[\delta||w]}$ when $\ell \equiv -\infty$ and $u \equiv +\infty$. The following technical claim shows that there exists an optimal deviation vector of special form.

Lemma 1 *Let $(S, \mathcal{F}, F^*, c, \ell, u, \|\cdot\|_{\infty,w})$ be a feasible minimum-cost inverse optimization problem and let p be an optimal deviation vector. Then $p_{[\delta|\ell,u|w]}$ is also an optimal deviation vector, where $\delta \coloneqq \max\{w(s) \cdot |p(s)| \mid s \in S\}$.*

Proof: The lower and upper bounds $\ell \leq p_{[\delta|\ell,u|w]} \leq u$ hold by definition, hence (b) is satisfied.

Now we show that (a) holds. The assumption $\ell \leq p \leq u$ and the definition of δ imply that $-\delta/w(s) \leq p(s) \leq u(s)$ and $\ell(s) \leq p(s) \leq \delta/w(s)$ hold for every $s \in S$. Let $F \in \mathcal{F}$ be an arbitrary solution. Then

$$\begin{aligned}
&(c - p_{[\delta|\ell,u|w]})(F^*) - (c - p_{[\delta|\ell,u|w]})(F) \\
&= \left[c(F^*) - \sum_{s \in F^*} p_{[\delta|\ell,u|w]}(s)\right] - \left[c(F) - \sum_{s \in F} p_{[\delta|\ell,u|w]}(s)\right] \\
&= c(F^*) - c(F) - \sum_{s \in F^* \setminus F} p_{[\delta|\ell,u|w]}(s) + \sum_{s \in F \setminus F^*} p_{[\delta|\ell,u|w]}(s) \\
&= c(F^*) - c(F) - \sum_{\substack{s \in F^* \setminus F \\ \ell(s) \leq \delta/w(s) \leq u(s)}} \delta/w(s) - \sum_{\substack{s \in F^* \setminus F \\ u(s) < \delta/w(s)}} u(s) + \sum_{\substack{s \in F \setminus F^* \\ -\delta/w(s) < \ell(s)}} \ell(s) + \sum_{\substack{s \in F \setminus F^* \\ \ell(s) \leq -\delta/w(s) \leq u(s)}} \big(-\delta/w(s)\big) \\
&\leq c(F^*) - c(F) - \sum_{s \in F^* \setminus F} p(s) + \sum_{s \in F \setminus F^*} p(s)
\end{aligned}$$

$$= \big(c(F^*) - p(F^*)\big) - \big(c(F) - p(F)\big)$$
$$= (c-p)(F^*) - (c-p)(F)$$
$$\leq 0,$$

where the last inequality holds by the feasibility of p.

Finally, to see that (c) holds for $p_{[\delta|\ell,u|w]}$, observe that $\|p_{[\delta|\ell,u|w]}\|_{\infty,w} \leq \delta = \|p\|_{\infty,w}$. This concludes the proof of the lemma. □

Corollary 2 *For any feasible minimum-cost inverse optimization problem $(S, \mathcal{F}, F^*, c, \ell, u, \|\cdot\|_{\infty,w})$, there exists $\delta \geq 0$ for which $p_{[\delta|\ell,u|w]}$ is an optimal deviation vector.*

2.1 Min-max characterization

With the help of Corollary 2, we are ready to provide a min-max characterization for the weighted infinity norm of an optimal deviation vector when no bounds are given.

Theorem 3 *Let $(S, \mathcal{F}, F^*, c, -\infty, +\infty, \|\cdot\|_{\infty,w})$ be a minimum-cost inverse optimization problem. Then*

$$\min\left\{ \|p\|_{\infty,w} \;\middle|\; p \text{ is a feasible deviation vector} \right\}$$
$$= \max\left\{ 0, \max\left\{ \frac{c(F^*) - c(F)}{\frac{1}{w}(F^* \triangle F)} \;\middle|\; F \in \mathcal{F},\ F \neq F^* \right\} \right\}.$$

Proof: Let p be an optimal deviation vector. By Corollary 2, we may assume that p is of the form $p_{[\delta||w]}$ for some $\delta \geq 0$. For ease of notation, let us define

$$d := \max\left\{ \frac{c(F^*) - c(F)}{\frac{1}{w}(F^* \triangle F)} \;\middle|\; F \in \mathcal{F},\ F \neq F^* \right\}.$$

If F^* is a minimum c-cost member of $\mathcal{F}$, then we are clearly done. Otherwise, $\delta, d > 0$ holds, and it suffices to show $\delta = d$. Let $F \in \mathcal{F}$, $F \neq F^*$ be arbitrary. Since $p_{[\delta||w]}$ is feasible, we get

$$0 \geq (c - p_{[\delta||w]})(F^*) - (c - p_{[\delta||w]})(F)$$
$$= \left[c(F^*) - \sum_{s \in F^*} \delta/w(s) \right] - \left[c(F) - \sum_{s \in F \cap F^*} \delta/w(s) - \sum_{s \in F \setminus F^*} \big(-\delta/w(s)\big) \right]$$
$$= c(F^*) - c(F) - \sum_{s \in F^* \triangle F} \delta/w(s)$$
$$= c(F^*) - c(F) - \delta \cdot \tfrac{1}{w}(F^* \triangle F).$$

This implies

$$\delta \geq \frac{c(F^*) - c(F)}{\frac{1}{w}(F^* \triangle F)},$$

hence $\delta \geq d$. To prove $\delta \leq d$, it is enough to show that $p_{[d||w]}$ is a feasible deviation vector. For any $F \in \mathcal{F}$, $F \neq F^*$, we have

$$\begin{aligned}(c - p_{[d||w]})(F^*) - (c - p_{[d||w]})(F) &= c(F^*) - c(F) - d \cdot \tfrac{1}{w}(F^* \triangle F) \\ &\leq c(F^*) - c(F) - \frac{c(F^*) - c(F)}{\frac{1}{w}(F^* \triangle F)} \cdot \tfrac{1}{w}(F^* \triangle F) \\ &= 0,\end{aligned}$$

which means that $p_{[d||w]}$ is indeed feasible. □

2.2 Algorithm

The goal of this section is to give a simple algorithm for determining an optimal deviation vector. First, we give a necessary and sufficient condition for the feasibility of the minimum-cost inverse optimization problem $(S, \mathcal{F}, F^*, c, \ell, u, \|\cdot\|_{\infty,w})$.

Lemma 4 *Let $(S, \mathcal{F}, F^*, c, \ell, u, \|\cdot\|_{\infty,w})$ be a minimum-cost inverse optimization problem. For any $F \in \mathcal{F}$, define*

$$W(F) \coloneqq \begin{cases} 1/\left(\displaystyle\sum_{\substack{s\in F^*\setminus F\\ u(s)=+\infty}} 1/w(s) + \sum_{\substack{s\in F\setminus F^*\\ \ell(s)=-\infty}} 1/w(s)\right) & \text{if the divisor is not 0,} \\ 0 & \text{otherwise,} \end{cases}$$

and

$$\begin{aligned} m_1 &\coloneqq \max\left\{w(s)\cdot|u(s)| \;\middle|\; s\in F^*, u(s)\neq+\infty\right\},\\ m_2 &\coloneqq \max\left\{w(s)\cdot|\ell(s)| \;\middle|\; s\in S\setminus F^*, \ell(s)\neq-\infty\right\},\\ m_3 &\coloneqq \max_{F\in\mathcal{F}}\left(W(F)\cdot\left(c(F^*)-c(F)-\sum_{\substack{s\in F^*\setminus F\\ u(s)\neq+\infty}} u(s)+\sum_{\substack{s\in F\setminus F^*\\ \ell(s)\neq-\infty}}\ell(s)\right)\right). \end{aligned}$$

Then the problem is feasible if and only if $p_{[m|\ell,u|w]}$ is a feasible deviation vector for

$$m \coloneqq \max\{0, m_1, m_2, m_3\}.$$

Proof: Clearly, if $p_{[m|\ell,u|w]}$ is feasible, then so is the problem.

To see the other direction, suppose to the contrary that $p_{[m|\ell,u|w]}$ is not feasible, but there exists a feasible deviation vector p. Then there exists $F \in \mathcal{F}$ such that

$$\begin{aligned} 0 &< (c-p_{[m|\ell,u|w]})(F^*) - (c-p_{[m|\ell,u|w]})(F)\\ &= c(F^*)-c(F) - \sum_{\substack{s\in F^*\setminus F\\ u(s)\neq+\infty}} u(s) - \sum_{\substack{s\in F^*\setminus F\\ u(s)=+\infty}} m/w(s) + \sum_{\substack{s\in F\setminus F^*\\ \ell(s)\neq-\infty}}\ell(s) + \sum_{\substack{s\in F\setminus F^*\\ \ell(s)=-\infty}}\big(-m/w(s)\big)\\ &= c(F)-c(F') - \sum_{\substack{s\in F^*\setminus F\\ u(s)\neq+\infty}} u(s) + \sum_{\substack{s\in F\setminus F^*\\ \ell(s)\neq-\infty}}\ell(s) - m\left(\sum_{\substack{s\in F^*\setminus F\\ u(s)=+\infty}} 1/w(s) + \sum_{\substack{s\in F\setminus F^*\\ \ell(s)=-\infty}} 1/w(s)\right). \end{aligned}$$

If $\{s\in F^*\setminus F \mid u(s)=+\infty\}\cup\{s\in F\setminus F^* \mid \ell(s)=-\infty\}=\emptyset$, then we obtain

$$\begin{aligned} 0 &< c(F^*)-c(F) - \sum_{\substack{s\in F^*\setminus F\\ u(s)\neq+\infty}} u(s) + \sum_{\substack{s\in F\setminus F^*\\ \ell(s)\neq-\infty}}\ell(s) - m\cdot 0\\ &\le c(F^*)-c(F) - \sum_{s\in F^*\setminus F} p(s) + \sum_{s\in F\setminus F^*} p(s)\\ &= \big(c(F^*)-p(F^*)\big)-\big(c(F)-p(F)\big)\\ &\le 0, \end{aligned}$$

where the last inequality holds since p is feasible, leading to a contradiction. If $\{s \in F^* \setminus F \mid u(s) = +\infty\} \cup \{s \in F \setminus F^* \mid \ell(s) = -\infty\} \neq \emptyset$, then we obtain

$$0 < c(F^*) - c(F) - \sum_{\substack{s \in F^* \setminus F \\ u(s) \neq +\infty}} u(s) + \sum_{\substack{s \in F \setminus F^* \\ \ell(s) \neq -\infty}} \ell(s) - m/W(F),$$

which contradicts the definition of m. □

Now we turn to the description of the algorithm and its analysis. The high-level idea is as follows. In each iteration, we determine an optimal solution $F \in \mathcal{F}$ of the underlying optimization problem using an oracle as a black box. If the cost of F equals that of F^*, then we stop. Otherwise, we modify the costs in such a way that F is 'eliminated', that is, F and F^* share the same cost with respect to the modified cost function – hence the name Newton-type algorithm. The algorithm is presented as Algorithm 1.

Algorithm 1: Algorithm for the minimum-cost inverse optimization problem under the weighted ℓ_∞-norm objective

Input: A minimum-cost inverse optimization problem $(S, \mathcal{F}, F^*, c, \ell, u, \|\cdot\|_{\infty,w})$ and an oracle $\mathcal{O}$ for the minimum-cost optimization problem $(S, \mathcal{F}, c')$ with any cost function c'.

Output: An optimal deviation vector if the problem is feasible, otherwise `Infeasible`.

$d_0 \leftarrow \max\left\{0, \max\{w(s) \cdot \ell(s) \mid s \in S, \ell(s) > 0\}, \max\left\{w(s) \cdot |u(s)| \;\middle|\; s \in S, u(s) < 0\right\}\right\}$
$c_0 \leftarrow c - p_{[d_0|\ell,u|w]}$
$F_0 \leftarrow$ a minimum c_0-cost member of $\mathcal{F}$ determined by $\mathcal{O}$
$i \leftarrow 0$
while $c_i(F^*) > c_i(F_i)$ **do**
  $S_i \leftarrow \{s \in F^* \mid d_i < w(s) \cdot u(s)\} \cup \{s \in S \setminus F^* \mid d_i > w(s) \cdot \ell(s)\}$
  if $(F^* \triangle F_i) \cap S_i \neq \emptyset$ **then**
    $\delta_{i+1} \leftarrow \min\left\{\frac{c_i(F^*) - c_i(F_i)}{\frac{1}{w}((F^* \triangle F_i) \cap S_i)}, \min_{s \in F^* \cap S_i}\left\{u(s) - \frac{d_i}{w(s)}\right\}, \min_{s \in (S \setminus F^*) \cap S_i}\left\{\frac{d_i}{w(s)} - \ell(s)\right\}\right\}$
  else
    return `Infeasible`
  $d_{i+1} \leftarrow d_i + \delta_{i+1}$
  $c_{i+1} \leftarrow c - p_{[d_{i+1}|\ell,u|w]}$
  $F_{i+1} \leftarrow$ a minimum c_{i+1}-cost member of $\mathcal{F}$ determined by $\mathcal{O}$
  $i \leftarrow i + 1$
return $p_{[d_i|\ell,u|w]}$

The correctness and the running time of the algorithm can be proved relying on the following lemmas.

Lemma 5 *For any $i \in \mathbb{N}$, if F^* is not a minimum c_i-cost member of $\mathcal{F}$, then either $\delta_{i+1} > 0$ or Algorithm 1 declares the problem to be infeasible.* □

Lemma 6 *For any $i \in \mathbb{N}$, if F^* is not a minimum c_i-cost member of $\mathcal{F}$, then either $S_{i+1} \subseteq S_i$ or Algorithm 1 declares the problem to be infeasible.* □

Lemma 7 *For any $i \in \mathbb{N}$, if F^* is not a minimum c_i-cost member of $\mathcal{F}$, then $c_{i+1}(F^*) = c_{i+1}(F_i)$ or $S_{i+1} \subsetneq S_i$, or Algorithm 1 declares the problem to be infeasible.* □

Lemma 8 *For any $i \in \mathbb{N}$, if F^* is not a minimum c_i-cost member of $\mathcal{F}$, then*

$$\tfrac{1}{w}((F_i - F^*) \cap S_i) - \tfrac{1}{w}((F_i \cap F^*) \cap S_i) > \tfrac{1}{w}((F_{i+1} - F^*) \cap S_{i+1}) - \tfrac{1}{w}((F_{i+1} \cap F^*) \cap S_{i+1})$$

or $S_{i+1} \subsetneq S_i$, or Algorithm 1 declares the problem to be infeasible. □

With the help of Lemmas 5-8, one can verify that Algorithm 1 solves the problem in pseudo-polynomial time, assuming that a pseudo-polynomial algorithm for the underlying optimization problem is available. However, if $w \equiv 1$ and the underlying optimization problem can be solved in (strongly) polynomial time for any cost function, then the algorithm finds an optimal deviation vector in (strongly) polynomial time.

Theorem 9 *Given a pseudo-polynomial algorithm for the minimum-cost optimization problem $(S, \mathcal{F}, c')$ for any cost function c', Algorithm 1 is a pseudo-polynomial-time algorithm for the minimum-cost inverse optimization problem $(S, \mathcal{F}, F^*, c, \ell, u, \|\cdot\|_{\infty,w})$.* □

3 Weighted span

Now we turn our attention to the weighted span objective. Recall that $w \in \mathbb{R}^S_+$ is a positive weight function. For any $\delta, \Delta \in \mathbb{R}$, let $p_{[\delta,\Delta|\ell,u|w]} \colon S \to \mathbb{R}$ be defined as

$$p_{[\delta,\Delta|\ell,u|w]}(s) \coloneqq \begin{cases} (\delta+\Delta)/w(s) & \text{if } s \in F^* \text{ and } \ell(s) \le (\delta+\Delta)/w(s) \le u(s), \\ \ell(s) & \text{if } s \in F^* \text{ and } (\delta+\Delta)/w(s) < \ell(s), \\ u(s) & \text{if } s \in F^* \text{ and } u(s) < (\delta+\Delta)/w(s), \\ \Delta/w(s) & \text{if } s \in S \setminus F^* \text{ and } \ell(s) \le \Delta/w(s) \le u(s), \\ \ell(s) & \text{if } s \in S \setminus F^* \text{ and } \Delta/w(s) < \ell(s), \\ u(s) & \text{if } s \in S \setminus F^* \text{ and } u(s) < \Delta/w(s). \end{cases}$$

We simply write $p_{[\delta,\Delta||w]}$ when $\ell \equiv -\infty$ and $u \equiv +\infty$. The following technical claim shows that there exists an optimal deviation vector of special form.

Lemma 10 *Let $\big(S, \mathcal{F}, F^*, c, \ell, u, \operatorname{span}_w(\cdot)\big)$ be a feasible minimum-cost inverse optimization problem and let p be an optimal deviation vector. Then $p_{[\delta,\Delta|\ell,u|w]}$ is also an optimal deviation vector, where $\Delta \coloneqq \min\big\{w(s)\cdot p(s) \;\big|\; s \in S\big\}$ and $\delta \coloneqq \max\big\{w(s)\cdot p(s) \;\big|\; s \in S\big\} - \Delta$.*

Proof: The lower and upper bounds $\ell \le p_{[\delta,\Delta|\ell,u|w]} \le u$ hold by definition, hence (b) is satisfied.

Now we show that (a) holds. The assumption $\ell \le p \le u$ and the definition of Δ and δ imply that $\ell(s) \le p(s) \le (\delta+\Delta)/w(s)$ and $\Delta/w(s) \le p(s) \le u(s)$ hold for every $s \in S$. Let $F \in \mathcal{F}$ be an arbitrary solution. Then

$$\begin{aligned}
&(c - p_{[\delta,\Delta|\ell,u|w]})(F^*) - (c - p_{[\delta,\Delta|\ell,u|w]})(F) \\
&= \left[c(F^*) - \sum_{\substack{s\in F^* \\ \ell(s)\le(\delta+\Delta)/w(s)\le u(s)}} (\delta+\Delta)/w(s) - \sum_{\substack{s\in F^* \\ u(s)<(\delta+\Delta)/w(s)}} u(s)\right] \\
&\quad - \left[c(F) - \sum_{\substack{s\in F\cap F^* \\ \ell(s)\le(\delta+\Delta)/w(s)\le u(s)}} (\delta+\Delta)/w(s) - \sum_{\substack{s\in F\cap F^* \\ u(s)<(\delta+\Delta)/w(s)}} u(s) - \sum_{\substack{s\in F\setminus F^* \\ \Delta/w(s)<\ell(s)}} \ell(s) - \sum_{\substack{s\in F\setminus F^* \\ \ell(s)\le\Delta/w(s)\le u(s)}} \Delta/w(s)\right] \\
&= c(F^*) - c(F) - \sum_{\substack{s\in F^*\setminus F \\ \ell(s)\le(\delta+\Delta)/w(s)\le u(s)}} (\delta+\Delta)/w(s) - \sum_{\substack{s\in F^*\setminus F \\ u(s)<(\delta+\Delta)/w(s)}} u(s) + \sum_{\substack{s\in F\setminus F^* \\ \Delta/w(s)<\ell(s)}} \ell(s) + \sum_{\substack{s\in F\setminus F^* \\ \ell(s)\le\Delta/w(s)\le u(s)}} \Delta/w(s) \\
&\le c(F^*) - c(F) - \sum_{s\in F^*\setminus F} p(s) + \sum_{s\in F\setminus F^*} p(s) \\
&= \big(c(F^*) - p(F^*)\big) - \big(c(F) - p(F)\big) \\
&\le 0,
\end{aligned}$$

where the last inequality holds by the feasibility of p.

Finally, to see that (c') holds for $p_{[\delta,\Delta|\ell,u|w]}$, observe that $\operatorname{span}_w(p) = \max\{w(s)\cdot p(s) \mid s\in S\} - \min\{w(s)\cdot p(s) \mid s\in S\} = \delta$ and $\operatorname{span}_w(p_{[\delta,\Delta|\ell,u|w]}) \le (\delta+\Delta)-\Delta = \delta$. That is, $p_{[\delta,\Delta|\ell,u|w]}$ is also optimal, concluding the proof of the lemma. □

Corollary 11 *For any feasible minimum-cost inverse optimization problem* $\big(S,\mathcal{F},F^*,c,\ell,u,\operatorname{span}_w(\cdot)\big)$, *there exist* $\delta,\Delta\in\mathbb{R}$ *for which* $p_{[\delta,\Delta|\ell,u|w]}$ *is an optimal deviation vector with*

$$\min\{w(s)\cdot p_{[\delta,\Delta|\ell,u|w]}(s) \mid s\in S\} = \Delta, \text{ and}$$
$$\max\{w(s)\cdot p_{[\delta,\Delta|\ell,u|w]}(s) \mid s\in S\} = \delta+\Delta.$$

Moreover,

$$\Delta \le \min\{w(s)\cdot u(s) \mid s\in S\}, \text{ and}$$
$$\delta+\Delta \ge \max\{w(s)\cdot \ell(s) \mid s\in S\}.$$

Proof: The first half is straightforward from Lemma 10. Since $\ell(s)\le p_{[\delta,\Delta|\ell,u|w]}(s)\le u(s)$ and $\Delta \le w(s)\cdot p_{[\delta,\Delta|\ell,u|w]}(s)\le \delta+\Delta$ hold for any $s\in S$, the second statement follows. □

3.1 Min-max characterization

With the help of Corollary 11, we are ready to provide a min-max characterization for the weighted span of an optimal deviation vector when no bounds are given.

Theorem 12 *Let* $\big(S,\mathcal{F},F^*,c,-\infty,+\infty,\operatorname{span}_w(\cdot)\big)$ *be a minimum-cost inverse optimization problem. Then*

$$\min\{\operatorname{span}_w(p) \mid p \text{ is a feasible deviation vector}\}$$
$$= \max\left\{0,\ \max\left\{\frac{c(F^*)-c(F'')}{\frac{1}{w}(F^*\setminus F'')}\,\middle|\, F''\in\mathcal{F},\ F''\neq F^*,\ \tfrac{1}{w}(F'')=\tfrac{1}{w}(F^*)\right\},\right.$$
$$\left.\max\left\{\frac{\frac{c(F^*)-c(F')}{\frac{1}{w}(F^*)-\frac{1}{w}(F')}-\frac{c(F^*)-c(F''')}{\frac{1}{w}(F^*)-\frac{1}{w}(F''')}}{\frac{\frac{1}{w}(F^*\setminus F')}{\frac{1}{w}(F^*)-\frac{1}{w}(F')}-\frac{\frac{1}{w}(F^*\setminus F''')}{\frac{1}{w}(F^*)-\frac{1}{w}(F''')}}\,\middle|\, F',F'''\in\mathcal{F},\ \tfrac{1}{w}(F')<\tfrac{1}{w}(F^*)<\tfrac{1}{w}(F''')\right\}\right\}.$$

Proof: Let p be an optimal deviation vector. By Corollary 11, we may assume that p is of the form $p_{[\delta,\Delta||w]}$ for some $\delta,\Delta\in\mathbb{R}$ such that $\min\{w(s)\cdot p(s) \mid s\in S\}=\Delta$ and $\max\{w(s)\cdot p(s) \mid s\in S\}=\delta+\Delta$. For ease of notation, let us denote the value of the maximum in the statement of the theorem by d. Furthermore,

$$D := \begin{cases} \max\limits_{\substack{F'\in\mathcal{F}\\ \frac{1}{w}(F')<\frac{1}{w}(F^*)}}\left\{\frac{c(F^*)-c(F')-d\cdot\frac{1}{w}(F^*\setminus F')}{\frac{1}{w}(F^*)-\frac{1}{w}(F')}\right\} & \text{if } \{F'\in\mathcal{F} \mid \tfrac{1}{w}(F')<\tfrac{1}{w}(F^*)\}\neq\emptyset,\\ \min\limits_{\substack{F'''\in\mathcal{F}\\ \frac{1}{w}(F''')>\frac{1}{w}(F^*)}}\left\{\frac{c(F^*)-c(F''')-d\cdot\frac{1}{w}(F^*\setminus F''')}{\frac{1}{w}(F^*)-\frac{1}{w}(F''')}\right\} & \text{if } \{F'\in\mathcal{F} \mid \tfrac{1}{w}(F')<\tfrac{1}{w}(F^*)\}=\emptyset \text{ and}\\ & \quad \{F'''\in\mathcal{F} \mid \tfrac{1}{w}(F''')>\tfrac{1}{w}(F^*)\}\neq\emptyset,\\ 0 & \text{otherwise.}\end{cases}$$

Define $\delta \coloneqq \max\left\{w(s)\cdot p(s) \;\middle|\; s\in S\right\} - \min\left\{w(s)\cdot p(s) \;\middle|\; s\in S\right\}$. Clearly, $\delta \geq 0$. Let $F \in \mathcal{F}$, $F \neq F^*$ be an arbitrary solution. Since $p_{[\delta,\Delta||w]}$ is feasible, we have

$$\begin{aligned}
0 &\geq (c - p_{[\delta,\Delta||w]})(F^*) - (c - p_{[\delta,\Delta||w]})(F) \\
&= \left(c(F^*) - \delta\cdot\tfrac{1}{w}(F^*) - \Delta\cdot\tfrac{1}{w}(F^*)\right) - \left(c(F^*) - \delta\cdot\tfrac{1}{w}(F\cap F^*) - \Delta\cdot\tfrac{1}{w}(F)\right) \\
&= c(F^*) - c(F) - \delta\cdot\tfrac{1}{w}(F^*\setminus F) - \Delta\cdot\left(\tfrac{1}{w}(F^*) - \tfrac{1}{w}(F)\right).
\end{aligned}$$

Thus for any $F'' \in \mathcal{F}$ such that $F'' \neq F^*$ and $\frac{1}{w}(F'') = \frac{1}{w}(F^*)$, if such F'' exists,

$$\delta \geq \frac{c(F^*) - c(F'')}{\frac{1}{w}(F^*\setminus F'')},$$

for any $F' \in \mathcal{F}$ such that $\frac{1}{w}(F') < \frac{1}{w}(F^*)$, if such F' exists,

$$\Delta \geq \frac{c(F^*) - c(F') - \delta\cdot\frac{1}{w}(F^*\setminus F')}{\frac{1}{w}(F^*) - \frac{1}{w}(F')},$$

and for any $F''' \in \mathcal{F}$ such that $\frac{1}{w}(F''') > \frac{1}{w}(F^*)$, if such F''' exists,

$$\Delta \leq \frac{c(F^*) - c(F''') - \delta\cdot\frac{1}{w}(F^*\setminus F''')}{\frac{1}{w}(F^*) - \frac{1}{w}(F''')}.$$

By the above, for any $F', F''' \in \mathcal{F}$ with $\frac{1}{w}(F') < \frac{1}{w}(F^*) < \frac{1}{w}(F''')$, if such F' and F''' exist, we have

$$\frac{c(F^*) - c(F') - \delta\cdot\frac{1}{w}(F^*\setminus F')}{\frac{1}{w}(F^*) - \frac{1}{w}(F')} \leq \frac{c(F^*) - c(F''') - \delta\cdot\frac{1}{w}(F^*\setminus F''')}{\frac{1}{w}(F^*) - \frac{1}{w}(F''')},$$

implying

$$\delta \geq \frac{\dfrac{c(F^*) - c(F')}{\frac{1}{w}(F^*) - \frac{1}{w}(F')} - \dfrac{c(F^*) - c(F''')}{\frac{1}{w}(F^*) - \frac{1}{w}(F''')}}{\dfrac{\frac{1}{w}(F^*\setminus F')}{\frac{1}{w}(F^*) - \frac{1}{w}(F')} - \dfrac{\frac{1}{w}(F^*\setminus F''')}{\frac{1}{w}(F^*) - \frac{1}{w}(F''')}}.$$

Therefore $\delta \geq d$ holds. To prove $\delta \leq d$, it is enough to show that $p_{[d,D||w]}$ is a feasible deviation vector. For any $F' \in \mathcal{F}$ with $\frac{1}{w}(F') < \frac{1}{w}(F^*)$, if such F' exists,

$$\begin{aligned}
&(c - p_{[d,D||w]})(F^*) - (c - p_{[d,D||w]})(F') \\
&\quad= c(F^*) - c(F') - d\cdot\tfrac{1}{w}(F^*\setminus F') - D\cdot\left(\tfrac{1}{w}(F^*) - \tfrac{1}{w}(F')\right) \\
&\quad\leq c(F^*) - c(F') - d\cdot\tfrac{1}{w}(F^*\setminus F') - \frac{c(F^*) - c(F') - d\cdot\frac{1}{w}(F^*\setminus F')}{\frac{1}{w}(F^*) - \frac{1}{w}(F')}\cdot\left(\tfrac{1}{w}(F^*) - \tfrac{1}{w}(F')\right) \\
&\quad= 0.
\end{aligned}$$

For any $F'' \in \mathcal{F}$ with $F'' \neq F$ and $\frac{1}{w}(F'') = \frac{1}{w}(F^*)$, if such F'' exists,

$$\begin{aligned}
&(c - p_{[d,D||w]})(F^*) - (c - p_{[d,D||w]})(F'') \\
&\quad= c(F^*) - c(F'') - d\cdot\tfrac{1}{w}(F^*\setminus F'') - D\cdot\left(\tfrac{1}{w}(F^*) - \tfrac{1}{w}(F'')\right) \\
&\quad= c(F^*) - c(F'') - d\cdot\tfrac{1}{w}(F^*\setminus F'') - D\cdot 0
\end{aligned}$$

$$\leq c(F^*) - c(F'') - \frac{c(F^*) - c(F'')}{\frac{1}{w}(F^* \setminus F'')} \cdot \tfrac{1}{w}(F^* \setminus F'')$$

$$= 0.$$

Let $F''' \in \mathcal{F}$ with $\frac{1}{w}(F''') > \frac{1}{w}(F^*)$ be arbitrary, if such F''' exists. First note that

$$D \leq \frac{c(F^*) - c(F''') - d \cdot \frac{1}{w}(F^* \setminus F''')}{\frac{1}{w}(F^*) - \frac{1}{w}(F''')}$$

holds since otherwise there would exist $F' \in \mathcal{F}$ with $\frac{1}{w}(F') < \frac{1}{w}(F^*)$ such that

$$\frac{c(F^*) - c(F') - d \cdot \frac{1}{w}(F^* \setminus F')}{\frac{1}{w}(F^*) - \frac{1}{w}(F')} > \frac{c(F^*) - c(F''') - d \cdot \frac{1}{w}(F^* \setminus F''')}{\frac{1}{w}(F^*) - \frac{1}{w}(F''')},$$

contradicting the definition of d. Thus,

$$\begin{aligned}
&(c - p_{[d,D||w]})(F^*) - (c - p_{[d,D||w]})(F''') \\
&= c(F^*) - c(F''') - d \cdot \tfrac{1}{w}(F^* \setminus F''') - D \cdot \left(\tfrac{1}{w}(F^*) - \tfrac{1}{w}(F''')\right) \\
&\leq c(F^*) - c(F''') - d \cdot \tfrac{1}{w}(F^* \setminus F''') - \frac{c(F^*) - c(F''') - d \cdot \frac{1}{w}(F^* \setminus F''')}{\frac{1}{w}(F^*) - \frac{1}{w}(F''')} \cdot \left(\tfrac{1}{w}(F^*) - \tfrac{1}{w}(F''')\right) \\
&= 0.
\end{aligned}$$

Therefore, $p_{[d,D||w]}$ is indeed feasible. □

3.2 Algorithm

Similarly to the case of the weighted ℓ_∞-norm, we give an algorithm for determining a feasible deviation vector with minimum weighted span. However, due to the different nature of the span objective, the algorithm and its analysis is significantly more complicated than the previous one.

Let us order the elements of the ground set $S = \{s_1, \ldots, s_n\}$ in such a way that $F^* = \{s_1, \ldots, s_{|F^*|}\}$. By Corollary 11, we may assume that

$$w(s_1) \cdot \ell(s_1) = \ldots = w\left(s_{|F^*|}\right) \cdot \ell\left(s_{|F^*|}\right) \geq w\left(s_{|F^*|+1}\right) \cdot \ell\left(s_{|F^*|+1}\right) \geq \ldots \geq w(s_n) \cdot \ell(s_n)$$

and

$$w(s_1) \cdot u(s_1) \geq \ldots \geq w\left(s_{|F^*|}\right) \cdot u\left(s_{|F^*|}\right) \geq w\left(s_{|F^*|+1}\right) \cdot u\left(s_{|F^*|+1}\right) = \ldots = w(s_n) \cdot u(s_n).$$

In the following, we look for the values of $\delta + \Delta$ and Δ for which $p_{[\delta,\Delta|\ell,u|w]}$ is an optimal deviation vector in the intervals

$$\begin{gathered}
\left[w\left(s_{|F^*|}\right) \cdot \ell\left(s_{|F^*|}\right),\ w\left(s_{|F^*|}\right) \cdot u\left(s_{|F^*|}\right)\right], \\
\left[w\left(s_{|F^*|}\right) \cdot u\left(s_{|F^*|}\right),\ w\left(s_{|F^*|-1}\right) \cdot u\left(s_{|F^*|-1}\right)\right], \\
\left[w\left(s_{|F^*|-1}\right) \cdot u\left(s_{|F^*|-1}\right),\ w\left(s_{|F^*|-2}\right) \cdot u\left(s_{|F^*|-2}\right)\right], \\
\vdots \\
\left[w(s_2) \cdot u(s_2),\ w(s_1) \cdot u(s_1)\right]
\end{gathered}$$

and

$$\Big[w\left(s_{|F^*|+1}\right)\cdot\ell\left(s_{|F^*|+1}\right),\ w\left(s_{|F^*|+1}\right)\cdot u\left(s_{|F^*|+1}\right)\Big],$$
$$\Big[w\left(s_{|F^*|+2}\right)\cdot\ell\left(s_{|F^*|+2}\right),\ w\left(s_{|F^*|+1}\right)\cdot\ell\left(s_{|F^*|+1}\right)\Big],$$
$$\Big[w\left(s_{|F^*|+3}\right)\cdot\ell\left(s_{|F^*|+3}\right),\ w\left(s_{|F^*|+2}\right)\cdot\ell\left(s_{|F^*|+2}\right)\Big],$$
$$\vdots$$
$$\Big[w(s_{n-1})\cdot\ell(s_{n-1}),\ w(s_n)\cdot\ell(s_n)\Big],$$

respectively. By the definition of $p_{[\delta,\Delta|\ell,u|w]}$, if $\delta+\Delta\in\big[w(s_{i+1})\cdot u(s_{i+1}), w(s_i)\cdot u(s_i)\big]$ for some $i\in\big[|F^*|-1\big]$, then $p_{[\delta,\Delta|\ell,u|w]}(s_j)=u(s_j)$ holds for all $j\in\big\{i+1,i+2,\ldots,|F^*|\big\}$. Similarly, if $\Delta\in\big[w(s_{i+1})\cdot\ell(s_{i+1}), w(s_i)\cdot\ell(s_i)\big]$ for some $i\in\big\{|F^*|+1,|F^*|+2,\ldots,n\big\}$, then $p_{[\delta,\Delta|\ell,u|w]}(s_j)=\ell(s_j)$ holds for all $j\in\big\{|F^*|+1,|F^*|+2,\ldots,i\big\}$. By the above and by Corollary 11, it is enough to consider the case when the lower and upper bounds are of the following form:

$$\ell(s)\coloneqq\begin{cases}0 & \text{if } s\in S_0,\\ \ell^{\text{in}}/w(s) & \text{if } s\in F^*-S_0,\\ \ell^{\text{out}}/w(s) & \text{otherwise}\end{cases}\qquad u(s)\coloneqq\begin{cases}0 & \text{if } s\in S_0,\\ u^{\text{in}}/w(s) & \text{if } s\in F^*-S_0,\\ u^{\text{out}}/w(s) & \text{otherwise}\end{cases}$$

$$\text{for some } S_0\subseteq S,\ \ell^{\text{in}},\ell^{\text{out}}\in\mathbb{R}\cup\{-\infty\} \text{ and } u^{\text{in}},u^{\text{out}}\in\mathbb{R}\cup\{+\infty\} \text{ satisfying}$$
$$\ell^{\text{in}}\le u^{\text{in}},\qquad \ell^{\text{out}}\le u^{\text{out}},\tag{SPEC-LU}$$
$$\max\big\{w(s)\ \big|\ s\in S-F^*\big\}\cdot\ell^{\text{out}}\le\min\big\{w(s)\ \big|\ s\in F^*\big\}\cdot\ell^{\text{in}},$$
$$\max\big\{w(s)\ \big|\ s\in S-F^*\big\}\cdot u^{\text{out}}\le\min\big\{w(s)\ \big|\ s\in F^*\big\}\cdot u^{\text{in}},$$
$$\ell^{\text{in}}\ge 0 \text{ if } S_0\cap F^*\neq\emptyset, \text{ and}$$
$$u^{\text{out}}\le 0 \text{ if } S_0\cap(S-F^*)\neq\emptyset.$$

Now we are ready to characterize the feasibility of a minimum-cost inverse optimization problem $\big(S,\mathcal{F},F^*,c,\ell,u,\text{span}_w(\cdot)\big)$.

Lemma 13 *Let $\big(S,\mathcal{F},F^*,c,\ell,u,\text{span}_w(\cdot)\big)$ be a minimum-cost inverse optimization problem, where ℓ and u satisfy (SPEC-LU). Let*

$$m_1\coloneqq\min\left\{\frac{c(F)-c(F^*)+u^{\text{in}}\cdot\frac{1}{w}\big((F^*-S_0)-(F-S_0)\big)}{\frac{1}{w}\big((F-S_0)-(F^*-S_0)\big)}\ \middle|\ F\in\mathcal{F},\ F-S_0\not\subseteq F^*-S_0\right\},$$
$$m_2\coloneqq\max\left\{\frac{c(F^*)-c(F)+\ell^{\text{out}}\cdot\frac{1}{w}\big((F-S_0)-(F^*-S_0)\big)}{\frac{1}{w}\big((F^*-S_0)-(F-S_0)\big)}\ \middle|\ F\in\mathcal{F},F^*-S_0\not\subseteq F-S_0\right\},$$
$$m_3\coloneqq\min\left\{\frac{c(F)-c(F^*)}{\frac{1}{w}\big((F-S_0)-(F^*-S_0)\big)}\ \middle|\ F\in\mathcal{F},F-S_0\not\subseteq F^*-S_0\right\},\ \textit{and}$$
$$m_4\coloneqq\max\left\{\frac{c(F^*)-c(F)}{\frac{1}{w}\big((F^*-S_0)-(F-S_0)\big)}\ \middle|\ F\in\mathcal{F},F^*-S_0\not\subseteq F-S_0\right\}.$$

- *If $u^{\text{in}}\neq+\infty$ and $\ell^{\text{out}}\neq-\infty$, then the minimum-cost inverse optimization problem is feasible if and only if $p_{[u^{\text{in}}-\ell^{\text{out}},\ell^{\text{out}}|\ell,u|w]}$ is a feasible deviation vector.*
- *If $u^{\text{in}}\neq+\infty$ and $\ell^{\text{out}}=-\infty$, then the minimum-cost inverse optimization problem is feasible if and only if $p_{[u^{\text{in}}-m,m|\ell,u|w]}$ is a feasible deviation vector, where*

$$m\coloneqq\begin{cases}m_1 & \textit{if } m_1\neq+\infty,\\ 0 & \textit{otherwise.}\end{cases}$$

- *If* $u^{\text{in}} = +\infty$ *and* $\ell^{\text{out}} \neq -\infty$*, then the minimum-cost inverse optimization problem is feasible if and only if* $p_{[M-\ell^{\text{out}},\ell^{\text{out}}|\ell,u|w]}$ *is a feasible deviation vector, where*

$$M \coloneqq \begin{cases} m_2 & \text{if } m_2 \neq -\infty, \\ 0 & \text{otherwise.} \end{cases}$$

- *If* $u^{\text{in}} = +\infty$ *and* $\ell^{\text{out}} = -\infty$*, then the minimum-cost inverse optimization problem is feasible if and only if* $p_{[M'-m',m'|\ell,u|w]}$ *is a feasible deviation vector, where*

$$m' \coloneqq \begin{cases} m_3 & \text{if } m_3 \neq +\infty, \\ 0 & \text{otherwise,} \end{cases} \quad \text{and} \quad M' \coloneqq \begin{cases} m_4 & \text{if } m_4 \neq -\infty, \\ 0 & \text{otherwise.} \end{cases}$$

Proof: Assume first that $u^{\text{in}} \neq +\infty$ and $\ell^{\text{out}} \neq -\infty$. Suppose to the contrary that $p_{[u^{\text{in}}-\ell^{\text{out}},\ell^{\text{out}}|\ell,u|w]}$ is not feasible, but there exists a feasible deviation vector p. Then there exists $F \in \mathcal{F}$ such that

$$\begin{aligned}
0 &< \big(c - p_{[u^{\text{in}}-\ell^{\text{out}},\ell^{\text{out}}|\ell,u|w]}\big)(F^*) - \big(c - p_{[u^{\text{in}}-\ell^{\text{out}},\ell^{\text{out}}|\ell,u|w]}\big)(F) \\
&= \left[c(F^*) - \sum_{s \in F^* - S_0} u^{\text{in}}/w(s)\right] - \left[c(F) - \sum_{s \in (F-S_0)\cap(F^*-S_0)} u^{\text{in}}/w(s) - \sum_{s \in (F-S_0)-(F^*-S_0)} \ell^{\text{out}}/w(s)\right] \\
&= c(F^*) - c(F) - \sum_{s \in (F^*-S_0)-(F-S_0)} u^{\text{in}}/w(s) + \sum_{s \in (F-S_0)-(F^*-S_0)} \ell^{\text{out}}/w(s) \\
&= c(F^*) - c(F) - \sum_{s \in (F^*-S_0)-(F-S_0)} u(s) + \sum_{s \in (F-S_0)-(F^*-S_0)} \ell(s) \\
&\leq c(F^*) - c(F) - \sum_{s \in (F^*-S_0)-(F-S_0)} p(s) + \sum_{s \in (F-S_0)-(F^*-S_0)} p(s) \\
&= \big[c(F^*) - p(F^*)\big] - \big[c(F) - p(F)\big] \\
&\leq 0,
\end{aligned}$$

a contradiction.

Now assume that $u^{\text{in}} \neq +\infty$ and $\ell^{\text{out}} = -\infty$. Suppose to the contrary that $p_{[u^{\text{in}}-m,m|\ell,u|w]}$ is not feasible but there exists a feasible deviation vector p. Then there exists $F \in \mathcal{F}$ such that

$$\begin{aligned}
0 &< \big(c - p_{[u^{\text{in}}-m,m|\ell,u|w]}\big)(F^*) - \big(c - p_{[u^{\text{in}}-m,m|\ell,u|w]}\big)(F) \\
&= c(F^*) - c(F) - \sum_{s \in (F^*-S_0)-(F-S_0)} u^{\text{in}}/w(s) + \sum_{s \in (F-S_0)-(F^*-S_0)} m/w(s) \\
&= c(F^*) - c(F) - u^{\text{in}} \cdot \tfrac{1}{w}\big((F^* - S_0) - (F - S_0)\big) + m \cdot \tfrac{1}{w}\big((F - S_0) - (F^* - S_0)\big).
\end{aligned}$$

The definition of m implies $F - S_0 \subseteq F^* - S_0$, otherwise the right-hand side of the above inequality should be non-positive. Thus

$$\begin{aligned}
0 &< c(F^*) - c(F) - u^{\text{in}} \cdot \tfrac{1}{w}\big((F^* - S_0) - (F - S_0)\big) + m \cdot \tfrac{1}{w}\big((F - S_0) - (F^* - S_0)\big) \\
&= c(F^*) - c(F) - u^{\text{in}} \cdot \tfrac{1}{w}\big((F^* - S_0) - (F - S_0)\big) + m \cdot 0 \\
&\leq (c-p)(F^*) - (c-p)(F) \\
&\leq 0,
\end{aligned}$$

a contradiction.

The remaining two cases can be proved analogously. □

With the help of Lemma 4 and a series of technical observations, one can give a pseudo-polynomial algorithm that determines a feasible deviation vector of minimum weighted span, assuming that a pseudo-polynomial algorithm for the underlying optimization problem is available. Similarly to the case of the weighted ℓ_∞-norm, if $w \equiv 1$ and the underlying optimization problem can be solved in (strongly) polynomial time for any cost function, then the algorithm finds an optimal deviation vector in (strongly) polynomial time.

Due to the high number of different cases discussed in the algorithm, we omit its description here, and only state the main result of the section.

Theorem 14 *Given a pseudo-polynomial algorithm for the minimum-cost optimization problem $(S, \mathcal{F}, c')$ for any cost function c', there is a pseudo-polynomial-time algorithm for the minimum-cost inverse optimization problem $\big(S, \mathcal{F}, F^*, c, \ell, u, \mathrm{span}_w(\cdot)\big)$.* □

References

[1] S. Ahmadian, U. Bhaskar, L. Sanità, and C. Swamy. Algorithms for inverse optimization problems. In *26th Annual European Symposium on Algorithms (ESA 2018)*, Leibniz International Proceedings in Informatics, LIPIcs. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2018.

[2] R. K. Ahuja and J. B. Orlin. Inverse optimization. *Operations Research*, 49(5):771–783, 2001.

[3] G. Bal. Inverse transport theory and applications. *Inverse Problems*, 25(5):053001, 2009.

[4] K. Bérczi, L. M. Mendoza-Cadena, and K. Varga. Inverse optimization problems with multiple weight functions. *Discrete Applied Mathematics*, 327:134–147, 2023.

[5] D. Burton and P. L. Toint. On an instance of the inverse shortest paths problem. *Mathematical Programming*, 53:45–61, 1992.

[6] T. C. Y. Chan, M. Eberg, K. Forster, C. Holloway, L. Ieraci, Y. Shalaby, and N. Yousefi. An inverse optimization approach to measuring clinical pathway concordance. *Management Science*, 68(3):1882–1903, 2021.

[7] M. Demange and J. Monnot. An introduction to inverse combinatorial problems. In *Paradigms of Combinatorial Optimization: Problems and New Approaches*, pages 547–586. John Wiley & Sons, Inc., second edition, 2014.

[8] Z. W. Di, S. Leyffer, and S. M. Wild. Optimization-based approach for joint X-ray fluorescence and transmission tomographic inversion. *SIAM Journal on Imaging Sciences*, 9(1):1–23, 2016.

[9] A. Frank and K. Murota. A discrete convex min-max formula for box-TDI polyhedra. *Mathematics of Operations Research*, 47(2):1026–1047, 2022.

[10] C. Heuberger. Inverse combinatorial optimization: A survey on problems, methods, and results. *Journal of Combinatorial Optimization*, 8(3):329–361, 2004.

[11] D. Jarrett and M. van der Schaar. Inverse active sensing: Modeling and understanding timely decision-making. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *PMLR*, pages 4713–4723, 2020.

[12] J. B. Lasserre. Inverse polynomial optimization. *Mathematics of Operations Research*, 38(3):418–436, 2013.

[13] Z. Liu and J. Zhang. On inverse problems of optimum perfect matching. *Journal of Combinatorial Optimization*, 7(3):215–228, 2003.

[14] M. Richter. *Inverse Problems: Basics, Theory and Applications in Geophysics*. Birkhäuser, 2016.

[15] Y. Xiaoguang. Note on inverse problem with ℓ_∞ objective function. *Applied Mathematics – A Journal of Chinese Universities*, 13(3):341–346, 1998.

[16] X. Yang and J. Zhang. Some inverse min-max network problems under weighted ℓ_1 and ℓ_∞ norms with bound constraints on changes. *Journal of Combinatorial Optimization*, 13(2):123–135, 2007.

[17] B. Zhang, X. Guan, and Q. Zhang. Inverse optimal value problem on minimum spanning tree under unit ℓ_∞ norm. *Optimization Letters*, 14(8):2301–2322, 2020.

[18] J. Zhang and Z. Liu. A further study on inverse linear programming problems. *Journal of Computational and Applied Mathematics*, 106(2):345–359, 1999.

[19] J. Zhang and Z. Liu. A general model of some inverse combinatorial optimization problems and its solution method under ℓ_∞ norm. *Journal of Combinatorial Optimization*, 6(2):207–227, 2002.

Supermodular Extension of Vizing's Edge-Coloring Theorem

Ryuhei Mizutani[1]

Department of Mathematical Informatics
The University of Tokyo
Tokyo, 113-8656, Japan
ryuhei_mizutani@mist.i.u-tokyo.ac.jp

Abstract: Kőnig's edge-coloring theorem for bipartite graphs and Vizing's edge-coloring theorem for general graphs are celebrated results in graph theory and combinatorial optimization. Schrijver generalized Kőnig's theorem to a framework defined with a pair of intersecting supermodular functions. The result is called the supermodular coloring theorem.

In this talk, we present a common generalization of Vizing's theorem and a weaker version of the supermodular coloring theorem. To describe this theorem, we introduce strongly triple-intersecting supermodular functions, which are extensions of intersecting supermodular functions.

Keywords: Edge-coloring, Supermodular coloring, Strongly triple-intersecting supermodular function

1 Introduction

Let $G = (V, E)$ be a multigraph. An *edge-coloring* of G is an assignment of colors to all edges in E such that no adjacent edges have the same color. The *edge-coloring number* $\chi'(G)$ of G is the minimum number k such that there exists an edge-coloring of G using k colors. For a vertex $v \in V$, the *degree* of v is the number of edges incident to v. Kőnig [4] showed the following relation between the edge-coloring number $\chi'(G)$ and the maximum degree $\Delta(G)$ of a bipartite multigraph G.

Theorem 1 (Kőnig [4]) *For a bipartite multigraph G, we have*

$$\chi'(G) = \Delta(G).$$

For any multigraph G, we have $\chi'(G) \geq \Delta(G)$ because the edges adjacent to the same vertex must have different colors. Theorem 1 states that this lower bound $\Delta(G)$ is equal to $\chi'(G)$ for every bipartite multigraph.

The *multiplicity* $\mu(G)$ of G is the maximum number of parallel edges in G. Vizing [8] showed the following analogue of Theorem 1 for general multigraphs.

Theorem 2 (Vizing [8]) *For any multigraph G, we have*

$$\Delta(G) \leq \chi'(G) \leq \Delta(G) + \mu(G).$$

Schrijver [6] extended Theorem 1 to a framework of supermodular functions on intersecting families. To describe this, we need some definitions. Let U be a finite set. A pair of $X, Y \subseteq U$ is called an *intersecting pair* if $X \cap Y \neq \emptyset$. A family $\mathcal{F} \subseteq 2^U$ is called an *intersecting family* if $X \cup Y, X \cap Y \in \mathcal{F}$ holds for every intersecting pair $X, Y \in \mathcal{F}$. A function $g : \mathcal{F} \to \mathbf{R}$ is called *intersecting supermodular* if $\mathcal{F}$ is an intersecting family and $g(X) + g(Y) \leq g(X \cup Y) + g(X \cap Y)$ holds for every intersecting pair $X, Y \in \mathcal{F}$. Schrijver [6] showed the following coloring-type theorem on an intersecting supermodular function.

[1]Research is supported by JST SPRING, Grant Number JPMJSP2108, Japan.

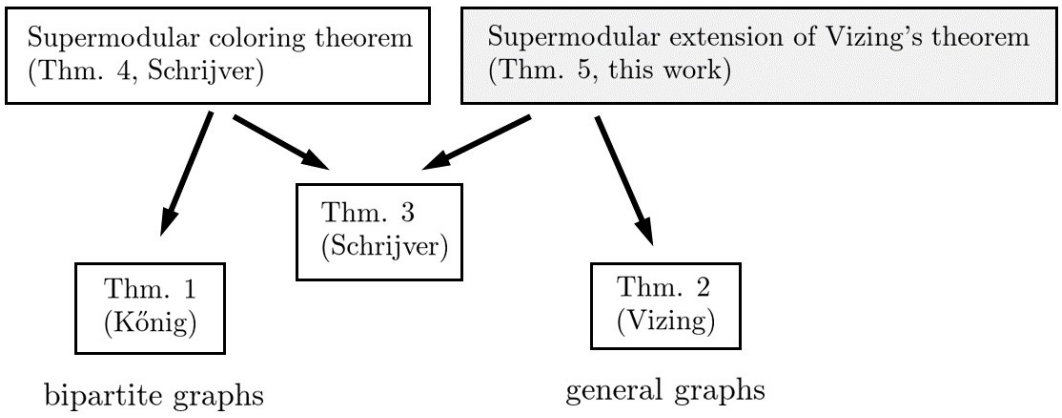


Figure 1: The relationship between the coloring-type theorems. The arrows mean implications.

Theorem 3 (Schrijver [6]) *Let $\mathcal{F} \subseteq 2^U$ be an intersecting family and $g : \mathcal{F} \to \mathbf{Z}$ an intersecting supermodular function. For $k \in \mathbf{Z}_{>0}$, if*

$$\min\{|X|, k\} \geq g(X)$$

holds for each $X \in \mathcal{F}$, then there exists an assignment of colors $\pi : U \to [k]$ satisfying

$$|\pi(X)| \geq g(X)$$

for each $X \in \mathcal{F}$, where $\pi(X) := \{\pi(u) \mid u \in X\}$ and $[k] := \{1, 2, \ldots, k\}$.

We are now ready to describe the *supermodular coloring theorem*, which is a generalization of Theorem 1 to the framework of intersecting supermodular functions.

Theorem 4 (Schrijver [6]) *Let $\mathcal{F}_1, \mathcal{F}_2 \subseteq 2^U$ be intersecting families, and $g_1 : \mathcal{F}_1 \to \mathbf{Z}$ and $g_2 : \mathcal{F}_2 \to \mathbf{Z}$ intersecting supermodular functions. For $k \in \mathbf{Z}_{>0}$, if*

$$\min\{|X|, k\} \geq g_i(X)$$

holds for each $i = 1, 2$ and each $X \in \mathcal{F}_i$, then there exists an assignment of colors $\pi : U \to [k]$ such that

$$|\pi(X)| \geq g_i(X)$$

holds for each $i = 1, 2$ and each $X \in \mathcal{F}_i$.

Tardos [7] gave an alternative proof of Theorem 4 using properties on generalized matroids. Theorem 4 was further extended to more general frameworks such as a framework including skew-supermodular coloring [2], and a framework of list supermodular coloring [3, 9].

Figure 1 describes the relationship between the above coloring-type theorems. A natural question arising from the supermodular coloring theorem is how to generalize Theorem 2 to a similar framework of supermodular functions.

2 Main result

Our main goal in this talk is to generalize Theorem 2 to a framework of a certain type of supermodular functions; i.e., we provide a common generalization of Theorems 2 and 3. To describe this, we need some definitions including new classes of intersecting families and intersecting supermodular functions. A

family $\mathcal{F} \subseteq 2^U$ is called a *strongly triple-intersecting family* if for every distinct $X_1, X_2, X_3 \in \mathcal{F}$ satisfying $X_1 \cap X_2 \cap X_3 \neq \emptyset$, we have

$$X_i \cup X_j, X_i \cap X_j \in \mathcal{F} \text{ and } X_k \cup X_l, X_k \cap X_l \in \mathcal{F}$$

for two pairs $(i,j),(k,l) \in \{(1,2),(2,3),(3,1)\}$. A function $g : \mathcal{F} \to \mathbf{R}$ is called *strongly triple-intersecting supermodular* if $\mathcal{F}$ is a strongly triple-intersecting family and for every distinct $X_1, X_2, X_3 \in \mathcal{F}$ satisfying $X_1 \cap X_2 \cap X_3 \neq \emptyset$, we have

$$X_i \cup X_j, X_i \cap X_j \in \mathcal{F} \text{ and } X_k \cup X_l, X_k \cap X_l \in \mathcal{F},$$

and

$$g(X_i) + g(X_j) \leq g(X_i \cup X_j) + g(X_i \cap X_j),$$
$$g(X_k) + g(X_l) \leq g(X_k \cup X_l) + g(X_k \cap X_l)$$

for two pairs $(i,j),(k,l) \in \{(1,2),(2,3),(3,1)\}$. For a family $\mathcal{F} \subseteq 2^U$ and a function $g : \mathcal{F} \to \mathbf{R}$, $\mathcal{L} \subseteq \mathcal{F}$ is called a *g-laminar family* if for every pair of sets $X, Y \in \mathcal{L}$, at least one of the following two conditions holds.

- At least one of $X \setminus Y, Y \setminus X, X \cap Y$ is the empty set.
- $X \cup Y, X \cap Y \in \mathcal{F}$ and $g(X) + g(Y) \leq g(X \cup Y) + g(X \cap Y)$ holds.

For $\mathcal{F} \subseteq 2^U$ and $X \in \mathcal{F}$, we denote $D_{\mathcal{F}}(X) := \max\{|X \cap Y| \mid Y \in \mathcal{F},\ X \not\subseteq Y \not\subseteq X\}$ (if $Y \in \mathcal{F}$ satisfying $X \not\subseteq Y \not\subseteq X$ does not exist, then we define $D_{\mathcal{F}}(X) := 0$). We are now ready to describe the common generalization of Theorems 2 and 3:

Theorem 5 *Let $\mathcal{F} \subseteq 2^U$ be a strongly triple-intersecting family and $g : \mathcal{F} \to \mathbf{Z}$ a strongly triple-intersecting supermodular function. For $k \in \mathbf{Z}_{>0}$, suppose that $\mathcal{L} := \{X \in \mathcal{F} \mid g(X) + D_{\mathcal{F}}(X) > k\}$ is a g-laminar family and*

$$\min\{|X|, k\} \geq g(X)$$

holds for every $X \in \mathcal{F}$. Then there exists an assignment of colors $\pi : U \to [k]$ such that

$$|\pi(X)| \geq g(X)$$

holds for every $X \in \mathcal{F}$.

The proof of this Theorem combines the proof technique of Theorem 2 by Berge and Fournier [1], and that of Theorems 3 and 4 by Schirijver [6]. See Figure 1 for the relationship between Theorem 5 and other coloring-type theorems. To see that Theorem 5 includes Theorem 2, let $\mathcal{F} := \{\delta(v) \mid v \in V\} \subseteq 2^E$ and let $g : \mathcal{F} \to \mathbf{Z}$ be a function defined by $g(X) := |X|$ for every $X \in \mathcal{F}$, where $\delta(v)$ denotes the set of edges incident to v. Then, $\mathcal{F}$ is a strongly triple-intersecting family because we have $\delta(v_1) \cap \delta(v_2) \cap \delta(v_3) = \emptyset$ for every distinct vertices $v_1, v_2, v_3 \in V$ (note that no edge in G has three endpoints v_1, v_2, v_3). This also implies that g is a strongly triple-intersecting supermodular function. Let $k := \Delta(G) + \mu(G)$. Then, $\mathcal{L} := \{\delta(v) \in \mathcal{F} \mid g(\delta(v)) + D_{\mathcal{F}}(\delta(v)) > \Delta(G) + \mu(G)\}$ is the empty set because we have $g(\delta(v)) = |\delta(v)| \leq \Delta(G)$ and $D_{\mathcal{F}}(\delta(v)) \leq \mu(G)$. Hence, $\mathcal{L}$ is a g-laminar family. We also have

$$\min\{|\delta(v)|, \Delta(G) + \mu(G)\} = |\delta(v)| = g(\delta(v))$$

for every $v \in V$. Therefore, by Theorem 5, there exists an assignment of colors $\pi : E \to [\Delta(G) + \mu(G)]$ such that $|\pi(\delta(v))| \geq g(\delta(v)) = |\delta(v)|$ holds for every $v \in V$, which implies Theorem 2.

Theorem 3 is also a special case of Theorem 5 as follows. Let $\mathcal{F}$ be an intersecting family, and g an intersecting supermodular function satisfying $\min\{|X|, k\} \geq g(X)$ for every $X \in \mathcal{F}$. Then $\mathcal{F}$ is also a strongly triple-intersecting family, and g is also a strongly triple-intersecting supermodular function. Moreover, $\mathcal{L} := \{X \in \mathcal{F} \mid g(X) + D_{\mathcal{F}}(X) > k\}$ is a g-laminar family because if $X, Y \in \mathcal{L}$ satisfy $X \cap Y \neq \emptyset$, then we have $X \cup Y, X \cap Y \in \mathcal{F}$ and $g(X) + g(Y) \leq g(X \cup Y) + g(X \cap Y)$. Hence, by Theorem 5, there exists an assignment of colors $\pi : U \to [k]$ such that $|\pi(X)| \geq g(X)$ holds for every $X \in \mathcal{F}$, which implies Theorem 3.

The original paper of this talk is available at arXiv preprint [5].

References

[1] C. Berge and J.-C. Fournier, A short proof for a generalization of Vizing's theorem, *Journal of Graph Theory* **15** (1991).

[2] A. Frank, T. Király, J. Pap, and D. Pritchard, Characterizing and recognizing generalized polymatroids, *Mathematical Programming* **146** (2014).

[3] S. Iwata and Y. Yokoi, List supermodular coloring, *Combinatorica* **38** (2018).

[4] D. Kőnig, Graphok és alkalmazásuk a determinánsok és a halmazok elméletére, *Mathematikai és Természettudományi Értesitő* **34** (1916).

[5] R. Mizutani, Supermodular extension of Vizing's edge-coloring theorem, *arXiv preprint arXiv:2211.07150*, 2022.

[6] A. Schrijver, Supermodular colourings, In *Matroid Theory* (L. Lovász and A. Recski, eds.), North-Holland, 1985.

[7] É. Tardos, Generalized matroids and supermodular colourings, In *Matroid Theory* (L. Lovász and A. Recski, eds.), North-Holland, 1985.

[8] V. G. Vizing, The chromatic class of a multigraph, *Cybernetics* **1** (1965).

[9] Y. Yokoi, List supermodular coloring with shorter lists, *Combinatorica* **39** (2019).

The extensible No-Three-In-Line problem

Dániel T. Nagy[1]

Department of Extremal Combinatorics
Alfréd Rényi Institute of Mathematics
Budapest, Hungary
nagydani@renyi.hu

Zoltán L. Nagy[2]

ELTE Linear Hypergraphs Research Group,
ELTE GAC Research Group
Eötvös Loránd University
Budapest, Hungary
nagyzoli@cs.elte.hu

Russ Woodroofe[3]

University of Primorska
Koper, Slovenia
russ.woodroofe@famnit.upr.si

Abstract: The classical No-Three-In-Line problem seeks the maximum number of points that may be selected from an $n \times n$ grid while avoiding a collinear triple. The maximum is well known to be linear in n. Following a question of Erde, we seek to select sets of large density from the infinite grid $\mathbb{Z}^2$ while avoiding a collinear triple. We show the existence of such a set which contains $\Theta(n/\log^{1+\varepsilon} n)$ points in $[1,n]^2$ for all n, where $\varepsilon > 0$ is an arbitrarily small real number. We also give computational evidence suggesting that a set of lattice points may exist that has at least $n/2$ points on every large enough $n \times n$ grid.

Keywords: no-three-in-line, collinear triples, square lattice

1 Introduction

A set of points in the plane are said to be in *general position* if no three of the points lie on a common line. Motivated by a problem concerning the placement of chess pieces, Dudeney [4] asked how many points may be placed in an $n \times n$ grid so that the points are in general position. This No-Three-In-Line problem has received considerable attention: for history and background, we refer to the book of Brass, Moser, and Pach [3] and that of Eppstein [5]; see also [10] for the problem in a higher dimensional setting. For an upper bound, it is straightforward to see that at most $2n$ points may so be placed. For rather small n, several examples have been constructed where the theoretical bound $2n$ can be attained [2, 6].

Joshua Erde proposed the following question at the Third Southwestern German Workshop on Graph Theory.

Problem 1 *Suppose that $S \subseteq \mathbb{Z}^2$ is a set of grid points in general position. Is it true that*

$$\liminf \frac{\left|S \cap [1,n]^2\right|}{n} = 0?$$

The purpose of this paper is to give evidence suggesting that the answer to the question may be "no".

[1]Research is supported by NKFIH grants FK 132060 and PD 137779 and by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences.

[2]Research is supported by NKFIH grants PD 134953 and K 124950.

[3]Research is supported by the Slovenian Research Agency (ARRS) research program P1-0285 and research projects J1-9108, N1-0160, J1-2451, and J3-3003.

While it is unknown for larger n whether the upper bound $2n$ is achievable in the $n \times n$ grid, there are several constructions where the size of the set is a smaller multiple of n. The earliest of these is due to Erdős (appearing in a paper published by Roth [11]), and uses the *modular parabola*, consisting of the points $(i, i^2) \mod p$. If $n = p$ is a prime number, then this yields n points in general position in $\mathbb{Z}^2 \cap [1, n]^2$. If n is not prime, then taking p to be the largest prime before n yields $n - o(n)$ points in general position.

The best known general construction for the No-Three-In-Line problem is due to Hall, Jackson, Sudbery, and Wild [7]. Their construction places points on a hyperbola $xy = k \pmod p$, where p is a prime slightly smaller than $n/2$, and yields $\frac{3}{2}n - o(n)$ points in general position.

Our aim in this paper is to give a bound on the growth rate of $|S \cap [1, n]^2|$. Thus, in contrast with the finite grid case, we need S to be dense in every square $[1, n]^2$ simultaneously.

A trivial lower bound follows from the parabola construction $\{(x, x^2) : x \in \mathbb{Z}^+\}$, giving $|S \cap [1, n]^2| = \Omega(n^{1/2})$. The constructions of Erdős and of Hall, Jackson, Sudbery, and Wild give large point sets in general position in any $n \times n$ grid, but rely heavily on choosing a prime based on n. These constructions do not straightforwardly generalize to an infinite set, as required for Question 1. Payne and Wood in [9] give a probabilistic construction (which can be turned into a probabilistic algorithm, as observed in [5, Algorithm 9.22]), but their techniques also rely on n being fixed. Thus, Problem 1 asks whether there are large sets of points in general position in the $n \times n$ grid, which can be extended nicely to larger sets of such points in larger grids.

Recently, Aichholzer, Eppstein, and Hainzl found the following lower bound on saturated subsets.

Theorem 2 (Aichholzer, Eppstein, and Hainzl [1]) *If S is a saturated subset of points in general position from an $n \times n$ grid, then $|S| = \Omega(n^{2/3})$.*

It follows straightforwardly that there is an infinite subset $S \subseteq \mathbb{Z}^2$ in general position so that $\left|S \cap [1, n]^2\right| = \Omega(n^{2/3})$.

2 Main result and proof idea

We show that the asymptotic growth of the size of the point set for the problem on the infinite grid can, in fact, be almost linear.

Theorem 3 *For any $\varepsilon > 0$, it is possible to construct a set $S \subseteq \mathbb{Z}^2$ of grid points in general position with*

$$\left|S \cap [1, n]^2\right| = \Theta(n/\log^{1+\varepsilon} n).$$

In particular, it holds that

$$\liminf \frac{\left|S \cap [1, n]^2\right|}{n/\log^{1+\varepsilon} n} > 0.$$

The main ingredients of the construction underlying Theorem 3 are as follows. We place separated copies of the parabola construction of Erdős along the curve $x/\log^{\varepsilon} x$. Specifically, for each value $x = 2^n$, we consider a square Q_n placed near the point $(2^n, 2^n/n^{\varepsilon})$ having side length a small multiple of $2^n/n^{1+\varepsilon}$. (See Figure 1.) We choose a suitable prime p_n for each integer n, and place a translated copy of Erdős' parabola construction with respect to p_n in the square. Finally, we delete those points from each such parabola that would form a collinear triple with points to their left.

By concavity, any line intersects the curve $x/\log^{\varepsilon} x$ in at most two points. If we choose the squares in the construction to be small enough, then (as we will see) a line intersects at most two of the squares. Thus to avoid collinear triples, it is enough to delete a point from each line which intersects the previously defined point set in one point and the parabola in the nth square in two points, or vice versa. By bounding from above the number of deleted points, we obtain $\Theta(\frac{N}{\log^{1+\varepsilon} N})$ lattice points in general position for each $[1, N]^2$, verifying Theorem 3.

The details of the proof are in the full version of this extended abstract [8].

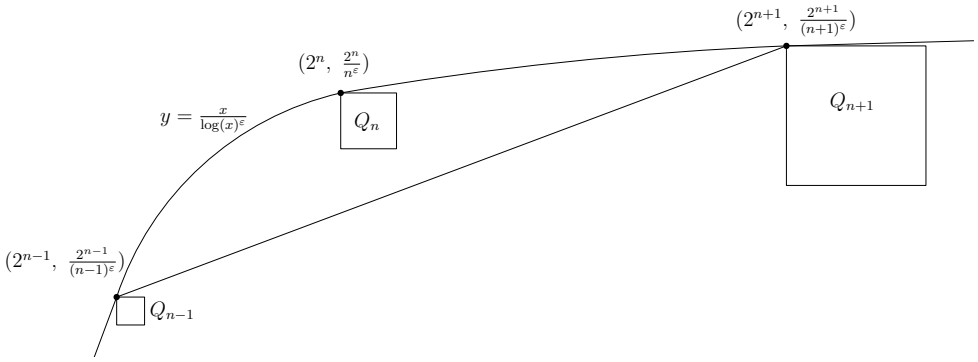


Figure 1: Illustration for the proof of Theorem 3 (not to scale)

3 Greedy lexicographic construction

A concrete and deterministic construction for large grid sets that are in general position may be described as follows. We iteratively look at each vertical line $x = 1$, $x = 2$, etc. At each line $x = i$, we find the least positive j so that the point (i, j) is not on a common line with any two previously selected points. We select this point (i, j) as well and move on to the next vertical line. This process yields an infinite set S_{lex} of triple-wise non-collinear integer points. This construction has been previously examined in OEIS sequence A236335, and similar constructions are the subject of sequences A236266, A179040.

Since we are interested in high density in squares $[1, n]^2$, we vary the lexicographic construction slightly to require $j < i$ at each step. If there is no allowed point (i, j) with $j < i$ for a given i, we move on to the next vertical line without selecting anything here. This process yields a set $S_{\text{lex}<}$. Experimental evidence suggests that $[1, n]^2 \cap S_{\text{lex}<}$ is of approximate size that is slightly larger than $0.8 \cdot n$. Refer to Table 1 for densities of $S_{\text{lex}<}$ at several values of n, or to Figure 2 for a plot of the first points in $S_{\text{lex}<}$.

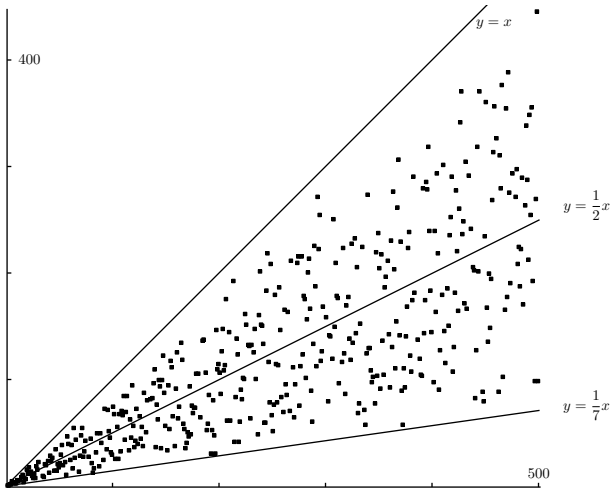


Figure 2: Construction $S_{\text{lex}<}$ for $n = 500$

We also consider the variant where we only take points on vertical lines having even x-intercept. In computer experiments, this even variant appears to find points without fail, yielding exactly n points in general position in $[1, 2n]^2$.

n	100	200	300	400	500	1000	2000	3000	4000	5000	10000
# points $S_{\text{lex}<}$	81	166	254	340	424	830	1678	2515	3353	4197	8385
density % n	81	83	84.6	85	84.8	83	83.9	83.8	83.8	83.9	83.9

Table 1: Density achieved by the lexicographic greedy construction

References

[1] O. Aichholzer, D. Eppstein, and E.-M. Hainzl, Geometric dominating sets - a minimum version of the No-Three-In-Line problem, *Computational Geometry* **108** (2023) Article 101913.

[2] D. B. Anderson, Update on the no-three-in-line problem, *J. Combinatorial Theory Ser. A* **27** (1979) 365–366.

[3] P. Brass, W. O. J. Moser, and J. Pach, Lattice point problems, Springer New York (2005) 417–433.

[4] H. E. Dudeney, Amusements in mathematics, Courier Corporation **473** (1917)

[5] D. Eppstein, Forbidden configurations in discrete geometry, Cambridge University Press (2018)

[6] A. Flammenkamp, Progress in the no-three-in-line problem, ii, *J. Combinatorial Theory Ser. A* **81** (1998) 108–113.

[7] R. R. Hall, T. H. Jackson, A. Sudbery, and K. Wild, Some advances in the no-three-in-line problem, *J. Combinatorial Theory Ser. A* **18** (1975) 336–341.

[8] D. T. Nagy, Z. L. Nagy and R. Woodroofe, The extensible No-Three-In-Line problem, arXiv:2209.01447 (2022)

[9] M. S. Payne and D. R. Wood, On the general position subset selection problem, *SIAM J. Discrete Math.* **27** (2013) 1727–1733.

[10] A. Pór and D. R. Wood, No-three-in-line-in-3D, *Algorithmica* **47** (2007) 481–488.

[11] K. F. Roth, On a problem of Heilbronn, *J. London Math. Soc.* **26** (1951) 198–204.

Simulations of quantum walks on regular graphs

Katalin Friedl

Department of Computer Science and Information Theory
Budapest University of Technology and Economics
Budapest, Hungary
friedl@cs.bme.hu

Viktória Nemkin

Department of Computer Science and Information Theory
Budapest University of Technology and Economics
Budapest, Hungary
nemkin@cs.bme.hu

Abstract: We have developed a simulator program in Python that can execute classical and quantum random walks on regular graphs. The user implements the oracle (a function that returns the adjacency list for a given vertex) and the quantum coin used in the simulation. The software simulates the walk and produces a Latex report file detailing the results. Running several simulations, we compared the behavior of classical and coined quantum walks on some regular graphs and demonstrated the periodicity in a few small special cases. We present such reports with some mathematical explanations.

Keywords: quantum walks, regular graphs, simulation

1 Introduction

Classical random walks are important tools for dealing with large instances of any computational problem that can be formulated as a search. They locally explore the space of possible solutions, iteratively improving on the current candidate via small transformations. It is not guaranteed that the global optimum will be found, however, they can discover good enough approximate solutions and are easy to implement, which can be useful in practical applications. For example, WalkSAT [6] is a popular random walk-based algorithm for testing the satisfiability of CNF formulas. It starts with a random truth assignment, then it repeatedly picks an unsatisfied clause and fixes it by flipping one of its variables until a solution is found or the iteration limit is reached.

Quantum walks [4, 7, 8] are generalized versions of classical random walks on a quantum computer. Since their introduction, the fact that their behaviour is different from their classical counterparts was demonstrated and referenced in many works [4, 5], and they are still being actively researched today. Grover's famous quantum search algorithm [3] can also be viewed as a special case of them[2, 1].

In the literature, there are two types of quantum random walks. The original coined version corresponds to the classical random walk that moves from vertex to vertex on a graph. Since this cannot be used for every graph, there is a more general (and somewhat more complicated) version of quantum walks due to Szegedy [7, 8]. Here we make experiments only with the first, coined version.

2 Quantum walks on regular graphs

In classical random walks on graphs, the only information stored about the system's state is the current position of the walker. This state is updated based on a random choice between the local outgoing edges.

One of the ways classical random walks can be formulated as a quantum algorithm is to implement the random choice as a quantum coin toss. We store the coin's state and update (toss) it using quantum

operators, which can result in a superposition of multiple states. The walker moves from a vertex on all of its outgoing edges in superposition (the current coin register determines the amplitudes). In general, the walker is in a superposition of vertices and moves to another superposition.

2.1 Quantum coins

For a d-regular graph, the current state of the coin is represented as a state vector in d-dimensional Hilbert space, each basis state corresponding to an outgoing edge choice. The coin toss is represented as a quantum operator, which is a $(d \times d)$ dimensional unitary matrix, acting on the state vector.

Based on the operator, several types of coins can be defined. The following ones are typically used in quantum walks.

Hadamard coin

The Hadamard coin is the most commonly used quantum coin. It is defined by the Hadamard-matrix, $\mathbf{H}^{\otimes n}$, where $\mathbf{H}$ is

$$\mathbf{H} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

An interesting property of this is that starting from one of the basis vector states, measuring after one toss results in a uniform random distribution while measuring after only the second consecutive toss results in the original state with 100% probability since $\mathbf{H}$ is Hermitian.

Grover coin

The Grover coin originates from Grover's search algorithm, where it is applied as the diffusion operator. Let $|D\rangle$ be the uniform state

$$|D\rangle = \mathbf{H}^{\otimes n} |0\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |i\rangle ,$$

then the Grover coin is the unitary matrix $\mathbf{G} = 2 |D\rangle \langle D| - \mathbf{I}$.

Fourier coin

In contrast to the Hadamard and Grover coins, the Fourier coin can be of any size, not just a power of 2. A Fourier-coin $\mathbf{F}_N$ of size N is defined by the matrix of the Quantum Fourier Transform

$$\mathbf{F}_N = \left[\frac{1}{\sqrt{N}} \omega^{xy} \right]_{x,y},$$

where $\omega = e^{\frac{2\pi i}{N}}$ is an Nth root of unity.

2.2 Quantum walk in 1 dimension

Following [4] the quantum walk on a line is defined using a particle characterised by its position $|x\rangle$ and coin (or spin) state $|s\rangle$. Actually, it is a walk on the circle but when the circle is large enough compared to the number of steps, this can be viewed as a quantum version of the classical random walk on the line.

Coin state In case of the circle, there are two directions to move so the coin state is represented in the two-dimensional space $\mathbb{C}^2$ with basis states $|0\rangle$ and $|1\rangle$. A coin state vector is a unit vector in the form $|s\rangle = s_0 |0\rangle + s_1 |1\rangle$.

Position state At the start of the walk, the particle is at the origin $|0\rangle$. After N steps a classical walker could be in any of the positions between $-N$ and N. For the quantum case, the position state is a unit vector in $\mathbb{C}^{2N+1}$, the basis vectors correspond to the possible positions, denoted by

$$|-N\rangle, |-N+1\rangle, \ldots, |-1\rangle, |0\rangle, |1\rangle, \ldots, |N-1\rangle, |N\rangle .$$

The position state vector is given by

$$|x\rangle = \sum_{i=-N}^{N} x_i |i\rangle .$$

Composite state The composite state of the system with the position and coin state is $|x\rangle \otimes |s\rangle$.

2.3 Evolution

The particle travels on the line based on its current coin state:

- If the current coin state is $|0\rangle$, the particle moves to the left, i.e. from position $|i\rangle$ to position $|i-1\rangle$.
- If the current coin state is $|1\rangle$, the particle moves to the right, i.e. from position $|i\rangle$ to position $|i+1\rangle$.

This step is realised with the unitary matrix $\mathbf{S}$ which operates on the complete state of the system, $|x\rangle \otimes |s\rangle$ and is assembled from a left and a right shift operator acting on $|x\rangle$ and another operator acting on $|s\rangle$ compiled using tensor product.

Left shift operator To move from position $|i\rangle$ to its left to $|i-1\rangle$ the position vector is multiplied with matrix $\mathbf{L}$ that can be expressed in the form

$$\mathbf{L} = |N\rangle \langle -N| + \sum_{i=-(N-1)}^{N} |i-1\rangle \langle i| .$$

It is easy to see that $\mathbf{L}|j\rangle = |j-1\rangle \langle j|j\rangle = |j-1\rangle$, when $-N < j \leq N$, while the first term on the right hand side closes the cycle, achieving $\mathbf{L}|-N\rangle = |N\rangle$, as it was desired.

Right shift operator Similarly, $\mathbf{R} = |-N\rangle \langle N| + \sum_{i=-N}^{N-1} |i+1\rangle \langle i|$ maps $|i\rangle$ to $|i+1\rangle$ and $|N\rangle$ to $|-N\rangle$ performing a right shift.

Shift operator Using matrices $\mathbf{L}$ and $\mathbf{R}$ operating on the position register $|x\rangle$ only, a unitary operator $\mathbf{S}$ can be defined, which operates on the composite state of the system, $|x\rangle \otimes |s\rangle$, executing matrix $\mathbf{L}$ on $|x\rangle$ only when $|s\rangle = |0\rangle$ and matrix $\mathbf{R}$ only when $|s\rangle = |1\rangle$,

$$\mathbf{S} = \mathbf{L} \otimes |0\rangle \langle 0| + \mathbf{R} \otimes |1\rangle \langle 1| .$$

The action of $\mathbf{S}$ on a vector $|x\rangle \otimes |s\rangle$ is $\mathbf{S}(|x\rangle \otimes |s\rangle) = s_0 |x-1, 0\rangle + s_1 |x+1, 1\rangle$.

So when the coin state is $|s\rangle = |0\rangle$ then this $\mathbf{S}$ maps $|x, 0\rangle$ to $|x-1, 0\rangle$ and in the case of $|s\rangle = |1\rangle$ $\mathbf{S}$ maps $|x, 1\rangle$ to $|x+1, 1\rangle$, as intended.

In the quantum setting the coin state can be any mixed state $s_0 |0\rangle + s_1 |1\rangle$ as well. In this case the particle will shift *both* to the left and to the right, at the same time. When measured, the particle can be found in position $|x-1\rangle$ with probability $|s_0|^2$ and in position $|x+1\rangle$ with probability $|s_1|^2$.

Coin operator To replace the classical coin tossing and to inject quantum superposition into the walk, the coin state can be transformed using an arbitrary 2 dimensional unitary matrix between the application of two shift operations. The Hadamard, Grover, and Fourier coins mentioned earlier are commonly used.

For any operator $\mathbf{C}$ on the coin register, the corresponding operator for the composite system that does not modify the position state is $\mathbf{I} \otimes \mathbf{C}$.

Evolution operator Combining the shift operator and the coin operator together, we obtain the following evolution operator, defining one step of the quantum walk on the line. The step consists of flipping the coin once, then applying the shifts

$$\mathbf{U} = \mathbf{S}(\mathbf{I} \otimes \mathbf{C}).$$

3 Quantum walk on regular graphs

Let us have an undirected, connected, regular graph. It will be useful to consider it also as a directed graph having the edges directed in both ways.

In a d-regular graph, the walker must choose from d possible edges to follow at every step. This suggests using a coin with d sides. In the quantum setting the previous 2-dimensional coin state is replaced by a d dimensional state vector, the basis vectors are $|0\rangle, |1\rangle, \ldots, |d-1\rangle$ corresponding to the different choices.

The coin operator is a unitary matrix $\mathbf{C} \in \mathbb{C}^{d \times d}$. The evolution operator formally looks the same as for the line, $\mathbf{U} = \mathbf{S}(\mathbf{I} \otimes \mathbf{C})$

To generalize the shift operator, the previous left and right shifts are replaced by d transition matrices with nonnegative elements,

$$\mathbf{S} = \mathbf{S}_0 \otimes |0\rangle \langle 0| + \mathbf{S}_1 \otimes |1\rangle \langle 1| + \cdots + \mathbf{S}_{d-1} \otimes |d-1\rangle \langle d-1|,$$

where $\mathbf{S}_0 + \mathbf{S}_1 + \cdots + \mathbf{S}_{d-1}$ is the adjacency matrix of the graph (as it was in the previous case, where $\mathbf{L} + \mathbf{R}$ was the adjacency matrix of the circle).

Since in a quantum walk the operators have to be unitary, $\mathbf{S}$ has to be unitary. This gives the following condition for the good decompositions of the adjacency matrix:

Theorem 1 *Let $\mathbf{S}_0, \ldots, \mathbf{S}_{d-1}$ be matrices with nonnegative elements and assume that $\sum_{i=0}^{d-1} \mathbf{S}_i$ is the adjacency matrix of a d-regular graph. The operator $\mathbf{S} = \sum_{i=0}^{d-1} \mathbf{S}_i \otimes |i\rangle \langle i|$ can be a shift operator of a quantum walk on the graph if and only if the $\mathbf{S}_i$ are permutation matrices.*

□

Corollary 2 *In the previous Theorem all the $\mathbf{S_i}$ are symmetric if and only if they correspond to a d coloring of the edges of the undirected graph, $\mathbf{S_i}$ is the adjacency matrix of the ith color class.*

□

Although usually this decomposition, based on edge coloring is mentioned, there are (nonsymmetric) possibilities, even when coloring the edges needs more than d (namely $d+1$) colors, since

Fact 3 *The adjacency matrix of any d-regular graph can be obtained as a sum of permutation matrices.*

4 Properties of quantum walks

Similarly to classical random walks, the effect of several steps in a quantum walk can be described by a power of the matrix representing one step. However, quantum walks behave differently than classical walks, since the matrix $\mathbf{U} = \mathbf{S}(I \otimes \mathbf{C})$ is unitary, therefore all of its eigenvalues have unit length, they cannot diminish. Furthermore, when all the eigenvalues are Mth roots of unity, then the walk is periodic by M.

The eigenvalues in some special cases can be computed from smaller matrices. Let $\lambda(\mathbf{A})$ denote the spectra of the operator $\mathbf{A}$. Then it is easy to see the following

Theorem 4 *Let* $\mathbf{U} = \mathbf{S}(I \otimes \mathbf{C})$ *where* $\mathbf{S} = \sum_{j=0}^{d-1} \mathbf{S}_j \otimes |j\rangle \langle j|$. *Assume that the* $\mathbf{S}_j$ *have a common eigenvector basis, i.e.* $\mathbf{S}_j v_k = \lambda_{j,k} v_k$, *where* $0 \leq j < d$ *and* $0 \leq k < n$. *Then* $\lambda(\mathbf{U}) = \bigcup_{k=0}^{n-1} \lambda(\mathbf{\Lambda}_k \mathbf{C})$, *where the* $\mathbf{\Lambda}_k$ *are* $d \times d$ *diagonal matrices formed from the* $\lambda_{j,k}$.

□

This shows that although the matrix of the walk has size $nd \times nd$ its eigenvalues can be computed from n matrices of sizes $d \times d$.

The condition of the Theorem is fulfilled, for example, when the $\mathbf{S}_j$ commute.

An interesting special case is when the graph is a Cayley graph of an Abelian group. Let Γ be an Abelian group, and $B \subseteq \Gamma$ be a symmetric generating system (i.e. $g \in B$ implies $g^{-1} \in B$). The vertices of the Cayley graph are the elements of Γ and there is an edge from a to b if $b = ag$ for some $g \in B$. This is a regular graph, the adjacency matrix of edges belonging to a fixed $g \in B$ form a permutation matrix, they provide a good decomposition for the adjacency matrix of the Cayley graph. It is known that they have common eigenvectors (formed from the characters of the group), the eigenvalues are values of characters.

Based on this, in some special cases, the eigenvalues of the quantum walk are not too difficult to compute.

For example, in the case of the circle when $n = 4$ then the eigenvalues are 8th roots of unity, when $n = 8$, then 24th roots of unity. This means that the walk is periodic in these cases. For a general n the eigenvalues $e^{i\varphi}$ are such that $\sqrt{2} \sin \varphi = \sin \frac{2\pi a}{n}$ holds ($a = 0, 1, \ldots, n-1$).

The rest of the section shows some results of our simulations.

4.1 Walks on a circle

The first walk to be reviewed is the 1-dimensional walk, it is a walk on a circle (with 128 vertices, numbered along the perimeter). Since this is a 2-regular graph, a 2-dimensional coin is used.

The 2-dimensional Hadamard and Fourier coins are identical, while the 2-dimensional Grover coin is just an X gate, which means the walker stays around to the starting position at all times.

In the following figures, we can see the changes in the probability distribution during the walk. The x axis contains the vertices, and the y axis contains the steps. The walker starts from the center (vertex 64), and in the classical case, multiple runs are done to arrive at a probability distribution, while in the quantum case, a single walker is enough, as it spreads in superposition over the graph.

The ballistic nature of the walk can be seen from steps 0 to around 100, where the bright red diagonals represent a strong probability concentration moving away from the origin. When the probability bumps reach the sides, they cross over and travel towards the center at the opposite side. We can also see secondary, tertiary, and further red lines traveling alongside the main ones. These reach cross over later, which results in a weaved pattern.

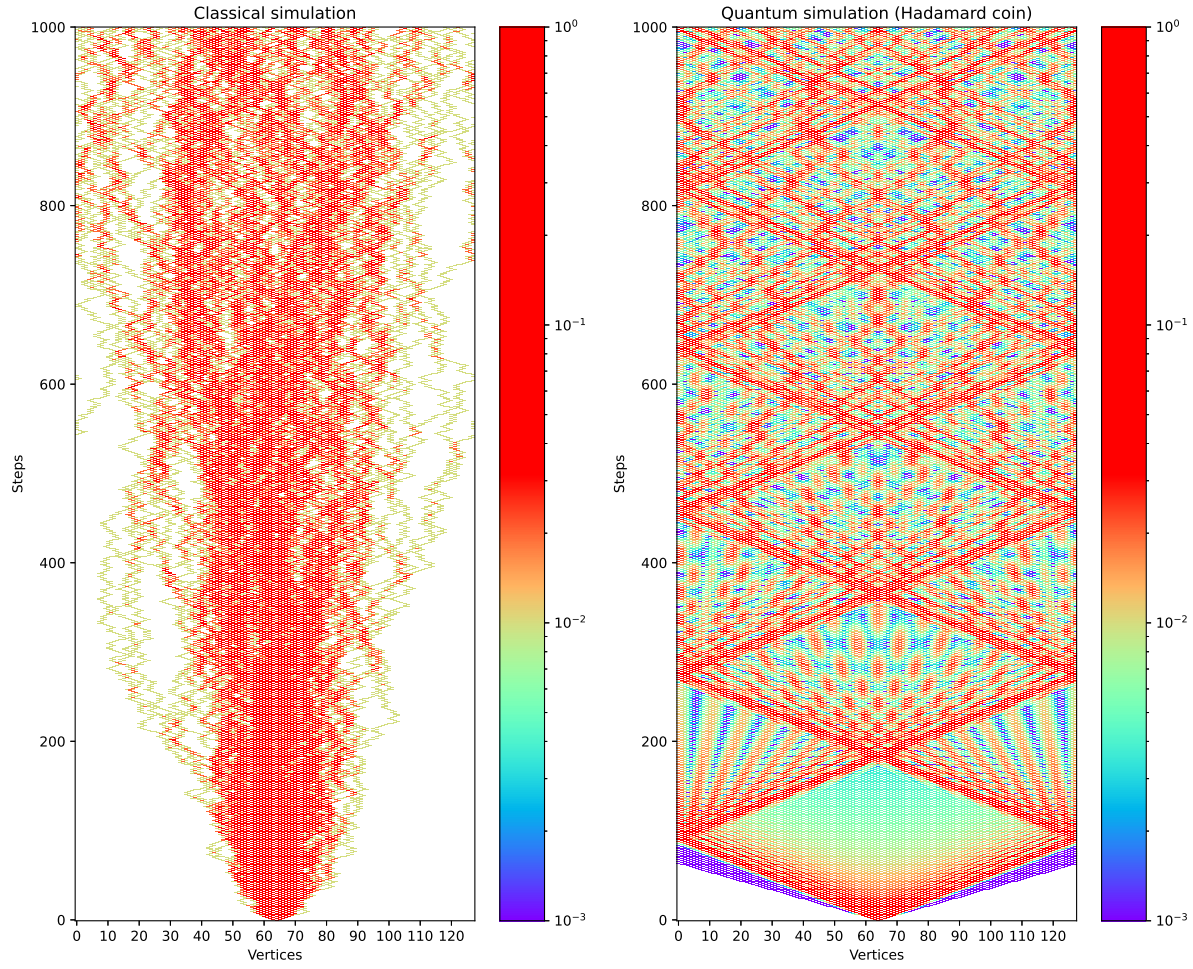


Figure 1: Probability distribution of walks on a larger circle of 128 vertices

4.2 Walks on the grid

The second graph reviewed is the 2-dimensional grid (torus), with $4 \times 4 = 16$ vertices, indexed in row-major order. The vertices are horizontally and vertically connected, the outer vertices connecting to the vertex at the opposite end of the same row or column.

The following 4 images contain the classical, the quantum Hadamard, the quantum Grover and the quantum Fourier walks on the grid. The walker starts from the center (vertex 8). The classical walk quickly spreads over the graph since all vertices are close to each other (as opposed to the line, where the maximum distance is large).

In the quantum case, using the Hadamard and Grover coins, an interesting quality can be distinctly observed: these quantum walks are periodic. On the 4×4 grid using the Hadamard coin, the periodicity is 40 steps, while using the Grover coin, it is 12 steps.

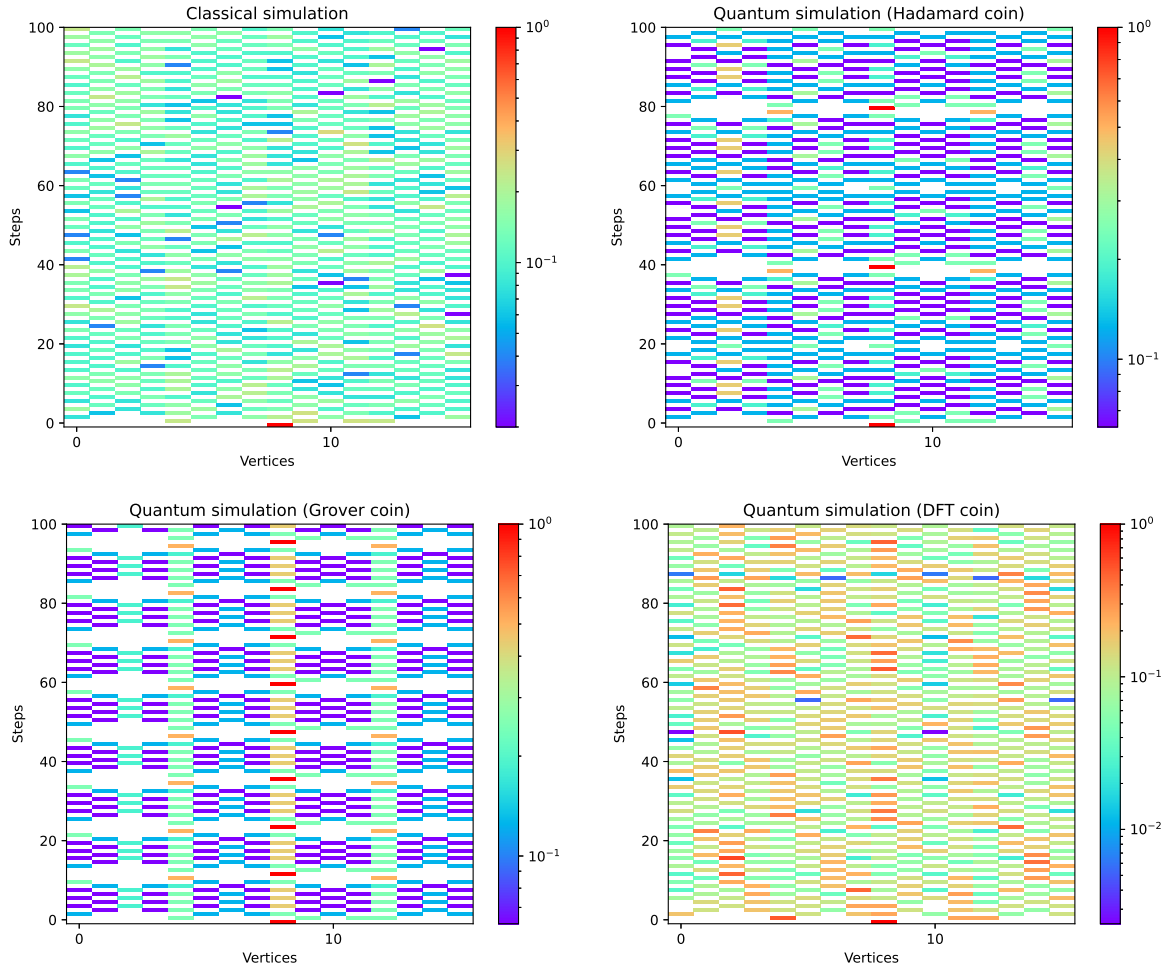


Figure 2: Probability distribution of walks on the grid (with horizontal / vertical steps)

4.3 Walks on hypercube

The third graph reviewed is the 4-dimensional boolean hypercube (with $2^4 = 16$ vertices).

On the 4 dimensional hypercube using the Hadamard coin, the periodicity is 24 steps, using the Grover coin, it is 12 steps while using the Fourier coin it turns out to be 48 steps.

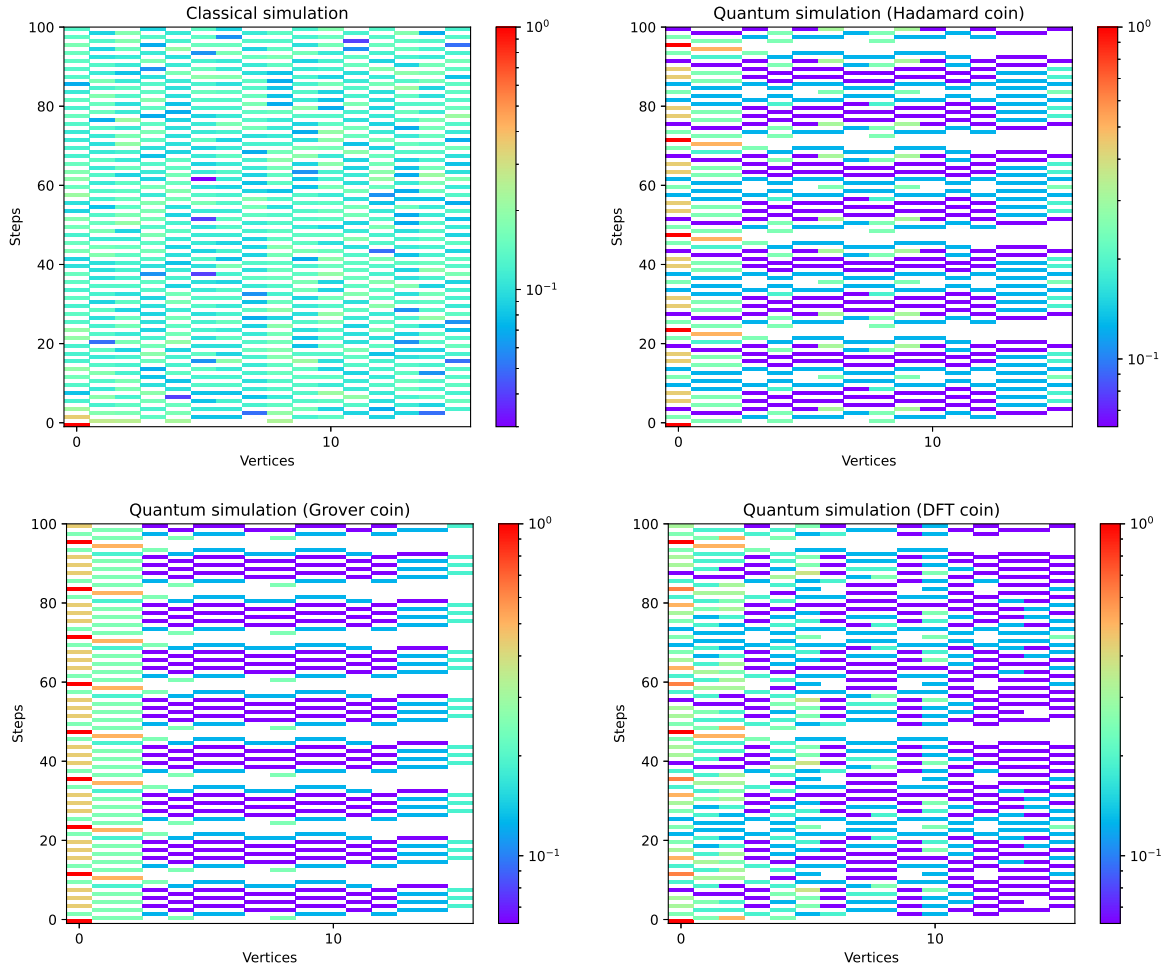


Figure 3: Probability distribution of walks on the hypercube

References

[1] A. Ambainis, Quantum walks and their algorithmic applications, *International Journal of Quantum Information* **1(4):507–518** (2003)

[2] F. Magniez, A. Nayak, J. Roland, and M. Santha, Search via Quantum Walk, *SIAM Journal on Computing* **40(1):142–164** (2011)

[3] L. K. Grover, A Fast Quantum Mechanical Algorithm for Database Search, *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing* **pp. 212–219** (1996)

[4] J. Kempe, Quantum random walks: An introductory overview, *Contemporary Physics* **44(4):307–327** (2003)

[5] R. Portugal, Quantum Walks and Search Algorithms, *Springer International Publishing* **Quantum Science and Technology** (2018)

[6] B. Selman, H. Kautz, and B. Cohen, Local Search Strategies for Satisfiability Testing, *Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge* **26** (1999)

[7] M. Szegedy, Quantum speed-up of Markov chain based algorithms, *45th Annual IEEE Symposium on Foundations of Computer Science* **pp. 32–41.** (2004)

[8] R. de Wolf, Quantum Computing: Lecture Notes, *arXiv:1907.09415* **v4** (2022)

Algebraic Algorithms for Fractional Linear Matroid Parity via Non-commutative Rank*

Taihei Oki†

Graduate School of Information Science and Technology
The University of Tokyo
Tokyo 113-8656, Japan
oki@mist.i.u-tokyo.ac.jp

Tasuku Soma

Department of Mathematics
Massachusetts Institute of Technology
Cambridge, MA 02139, United States
tasuku@mit.edu

Abstract: Matrix representations are a powerful tool for designing efficient algorithms for combinatorial optimization problems such as matching, and linear matroid intersection and parity. In this paper, we initiate the study of matrix representations using the concept of non-commutative rank (nc-rank), which has recently attracted attention in the research of Edmonds' problem. We reveal that the nc-rank of the matrix representation of linear matroid parity corresponds to the optimal value of fractional linear matroid parity: a half-integral relaxation of linear matroid parity. Based on our representation, we present an algebraic algorithm for the fractional linear matroid parity problem by building a new technique to incorporate the search-to-decision reduction into the half-integral problem represented via the nc-rank. Our algorithm is significantly simpler and faster than the existing algorithm.

Keywords: fractional matching, fractional matroid parity, non-commutative Edmonds' problem, non-commutative rank, search-to-decision reduction

1 Introduction

Matrix representations of combinatorial optimization problems have been a powerful tool for designing efficient algorithms. This line of research originates in Tutte's work [21] for the matching problem, where his matrix representation is now known as the *Tutte matrix*. Edmonds [6] dealt with a simpler representation for the bipartite case, and this result was later extended by Tomizawa and Iri [20] to the linear matroid intersection problem. Unifying these works, Lovász [17] gave a matrix representation for the *linear matroid parity problem* [16] (also called the *linear matroid matching problem*), a common generalization of the matching and linear matroid intersection problems.

These matrix representations are of the form so called *linear* (*symbolic*) *matrices*:

$$A = \sum_{i=1}^{m} x_i A_i, \tag{1}$$

where $x_1, \dots, x_m$ are distinct indeterminates (symbols), and $A_1, \dots, A_m$ are constant matrices over a field $\mathbb{K}$ determined from a given instance of the problem. Intuitively, each x_i corresponds to each element of the problem's ground set, and setting x_i to zero means removing the element from consideration. Every A_i is (i) a matrix having only one nonzero entry for the bipartite matching problem, (ii) a skew-symmetric matrix having two nonzero entries for the matching problem, (iii) a rank-one matrix for the linear matroid intersection problem, and (iv) a rank-two skew-symmetric matrix for the linear matroid parity problem. The rank of A (as a matrix over the rational function field $\mathbb{K}(x_1, \dots, x_m)$) coincides with

*The full version of this paper is available at https://arxiv.org/abs/2207.07946.

†Research is supported by JSPS KAKENHI Grant Number JP22K17853 and JST ERATO Grant Number JPMJER1903.

the size of maximum bipartite matchings and maximum common independent sets, and with twice the size of maximum matching and maximum independent parity set.

The problem of computing the rank of a given linear matrix is called *Edmonds' problem* [6]. When $|\mathbb{K}|$ is large enough, we can solve Edmonds' problem by substituting random values drawn from $\mathbb{K}$ into the indeterminates $x_1, \ldots, x_m$ and computing the rank of the obtained constant matrix; the probability of success can be bounded by the Schwartz–Zippel lemma [19, 23]. This remarkably simple idea leads to efficient randomized polynomial-time algorithms for various combinatorial optimization problems via matrix representations. These algorithms are called *algebraic algorithms*. Indeed, the current fastest algorithm for linear matroid intersection [11] and parity [4] are algebraic algorithms.

A major open question in Edmonds' problem is to develop a *deterministic* polynomial-time algorithm for general A_i; the existence of such an algorithm would imply non-trivial circuit complexity lower bounds [15]. To shed light on this question, recent studies [8, 10, 12, 14] focused on the *non-commutative* version of Edmonds' problem. This problem is to compute the *non-commutative rank* (nc-rank) of a linear matrix: the rank when the indeterminates $x_1, \ldots, x_m$ are regarded as "pairwise non-commutative", i.e., $x_i x_j \neq x_j x_i$ if $i \neq j$. An equivalent and elementary definition of nc-rank involves the *blow-up* of linear matrices. For $d \geq 1$, the dth-order *blow-up* of an $n \times n$ linear matrix A is a $dn \times dn$ linear matrix

$$A^{\{d\}} = \sum_{i=1}^{m} X_i \otimes A_i, \tag{2}$$

where $X_1, \ldots, X_m$ are $d \times d$ matrices of distinct indeterminates in their entries, and $\otimes$ denotes the Kronecker product. Then, nc-rank A is equal to $\frac{1}{d} \operatorname{rank} A^{\{d\}}$ for $d \geq n-1$ [5]. The nc-rank can also be defined via a min-max type formulation [7]. Recent breakthrough results [8, 10, 14] show that non-commutative Edmonds' problem is solvable in deterministic polynomial time.

Given the recent advances in the studies of nc-rank, it is quite natural to ask: *Can we devise an efficient and simple randomized algorithm for combinatorial optimization problems via nc-rank?* In this paper, we initiate the study of algebraic algorithms via nc-rank. We first reveal that the nc-rank of the matrix representation of the linear matroid parity problem is equal to twice the optimal value of the corresponding *fractional linear matroid parity problem*, which is a continuous relaxation of the linear matroid parity problem introduced by Vande Vate [22]. The set of feasible solutions of this problem, called the *fractional matroid parity polytope*, is a half-integral polytope contained in the 0-1 hypercube whose integral points correspond to the feasible solutions of the linear matroid parity problem. We prove our claim by establishing a correspondence between dual solutions of the fractional linear matroid parity and non-commutative Edmonds' problems.

This result provides a "matrix representation" that involves the nc-rank for the fractional linear matroid parity problem. Thus, we can compute the optimal value of fractional linear matroid parity by any deterministic polynomial-time algorithms [8, 10, 14] for non-commutative Edmonds' problem, and if $|\mathbb{K}|$ is large, one can also use the simple randomized algorithm that substitutes random values into the entries of $X_1, \ldots, X_m$ in $A^{\{n-1\}}$ with high probability, where n is the dimension of the space. These algorithms, however, do not output an actual optimal *solution*. In the known matrix representations of matching and linear matroid intersection and parity, this issue can be addressed with the *search-to-decision* reduction: we can test whether each element, say i, is contained in an optimal solution by simply checking whether setting $x_i = 0$ decreases the rank.

To incorporate the search-to-decision reduction into fractional linear matroid parity, whose extreme solutions are half-integral, we establish a novel correspondence between the rank of X_i in the blow-up and the ith component of a solution. Letting A be the matrix representation of this problem, we prove that nc-rank $A = \frac{1}{2} \operatorname{rank} A^{\{2\}}$ holds, i.e., the second-order blow-up is enough for attaining the nc-rank. Then, roughly speaking, we show that restricting the rank of X_i to 0, 1, or 2 corresponds to setting an upper bound on the ith component y_i of a solution $y \in \mathbb{R}^m$ to 0, $\frac{1}{2}$, or 1, respectively. Our algorithm runs in $O(n^\omega + mn^2)$ time, where n is the dimension of the space, m is the size of the ground set (see Section 2 for precise definitions), and $2 \leq \omega \leq 3$ is the matrix multiplication exponent. This is faster than the existing algorithm of Chang et al. [2] that takes $O(m^4 n^\omega)$ time.

In the full version of this paper, we further present a much faster divide-and-conquer algorithm that runs in $O(mn^{\omega-1})$ time. The full paper also develops an algebraic algorithm to obtain a dual optimal solution. Interested readers are referred to the arXiv preprint.

2 Preliminaries

We give basic definitions and notations. Let $\mathbb{N}$ be the set of natural numbers and $\mathbb{R}$ the set of reals. For $n \in \mathbb{N}$, let $[n] \coloneqq \{1, 2, \ldots, n\}$. For two real vectors $y = (y_1, \ldots, y_m)$ and $z = (z_1, \ldots, z_m)$, $y \leq z$ means that $y_i \leq z_i$ for all $i \in [m]$. The *cardinality* of a nonnegative vector $y \in \mathbb{R}^m$ is $|y| \coloneqq \sum_{i=1}^m y_i$. Let $\mathbf{0}$ and $\mathbf{1}$ denote the all-zero and all-one vectors, respectively, of appropriate dimensions. Let e_i be the ith standard unit vector, i.e., its jth component is 1 if $i = j$ and 0 otherwise.

Let $\mathbb{K}$ be a ground field. We assume that arithmetic operations on $\mathbb{K}$ can be performed in constant time. We denote by $\mathrm{GL}_n(\mathbb{K})$ the set of $n \times n$ nonsingular matrices over $\mathbb{K}$. For a matrix A, a row subset I, and a column subset J, we denote by $A[I, J]$ the submatrix indexed by I and J, and by $A[I]$ the principal submatrix $A[I, I]$ for square A. When I (resp. J) is all the rows (resp. columns), we denote $A[I, J]$ by $A[*, J]$ (resp. $A[I, *]$). The $n \times n$ identity matrix and the $n \times m$ zero matrix are denoted as I_n and $O_{n,m}$, respectively. We will omit the subscript of a zero matrix when its size does not matter.

A square matrix $A \in \mathbb{K}^{n \times n}$ is said to be *skew-symmetric* if $A^\top = -A$ and its diagonals are zero. For two vectors $a, b \in \mathbb{K}^n$, we define the *wedge product* as $a \wedge b \coloneqq ab^\top - ba^\top$. This is a skew-symmetric matrix of rank-two if a and b are linearly independent. For $V, W \subseteq \mathbb{K}^n$, we mean by $V \leq W$ that V and W are vector subspaces of $\mathbb{K}^n$ such that $V \subseteq W$, i.e., V is a subspace of W, and $V < W$ means $V \leq W$ and $V \neq W$. For vectors $a_1, \ldots, a_m \in \mathbb{K}^n$, let $\langle a_1, \ldots, a_m \rangle$ denote the vector subspace spanned by $a_1, \ldots, a_m$.

2.1 Linear Matroid Parity and Fractional Linear Matroid Parity

Let $\ell_1, \ldots, \ell_m \leq \mathbb{K}^n$ be two-dimensional vector subspaces, called *lines*. A line subset $M \subseteq L \coloneqq \{\ell_1, \ldots, \ell_m\}$ is called a *matroid matching* if it spans a $2|M|$-dimensional vector subspace of $\mathbb{K}^n$. A *parity base* is a matroid matching M with $2|M| = n$. Without loss of generality, we assume $n \leq 2m$ since we can focus on the at most $2m$-dimensional subspace spanned by the lines. The *linear matroid parity problem* [16] (or the *linear matroid matching problem*) is to find a matroid matching of maximum cardinality. For the linear matroid parity problem, Lovász [17] introduced the following matrix representation:

$$A = \sum_{i=1}^{m} x_i (a_i \wedge b_i), \tag{3}$$

where $\{a_i, b_i\}$ is any basis of ℓ_i for $i \in [m]$ and $x_1, \ldots, x_m$ are distinct indeterminates. That is, A is a linear matrix (1) with rank-two skew-symmetric coefficients.

Theorem 1 ([17, 18]) *Let A be the matrix representation* (3) *corresponding to lines L. Then, we have*

$$\operatorname{rank} A = 2 \max\{|M| : M \subseteq L \text{ is a matroid matching}\}.$$

The *matroid parity polytope* is the convex hull of the incidence vectors of the matroid matchings. In contrast to the matching and matroid intersection polytopes, a polyhedral description of matroid parity polytopes is still unknown. As a relaxation of the matroid parity polytope, Vande Vate [22] introduced a *fractional matroid parity (matching) polytope* as follows. Let $L = \{\ell_1, \ldots, \ell_m\}$ be lines. A *fractional matroid matching* is a nonnegative vector $y \in \mathbb{R}^m$ such that $\sum_{i=1}^m \dim(S \cap \ell_i) y_i \leq \dim S$ holds for all $S \leq \mathbb{K}^n$. The *fractional matroid parity polytope* P is the set of all fractional matroid matchings. This polytope is half-integral, i.e., extreme fractional matroid matchings are half-integral, and the integral ones are the incidence vectors of the matroid matchings [22].

The *fractional linear matroid parity (matching) problem* is to find a fractional matroid matching of maximum cardinality. Since fractional matroid parity polytopes are half-integral, there always exists

a half-integral optimal solution. Chang et al. [2, 3] gave a min-max theorem and a polynomial-time algorithm for this problem. We shall define a *2-cover* as a pair (S, T) of vector subspaces of $\mathbb{K}^n$ such that $\dim(S \cap \ell_i) + \dim(T \cap \ell_i) \geq 2$ for all $i \in [m]$. A 2-cover (S, T) is said to be *nested* if $S \leq T$.

Theorem 2 ([2, Corollary 4.3]) *For a fractional matroid parity polytope P, it holds*

$$2 \max_{y \in P} |y| = \min_{(S,T): \text{nested 2-cover}} (\dim S + \dim T). \tag{4}$$

A minimizer in (4) is called a *minimum 2-cover*. By the modularity of the dimension, if (S, T) and (S', T') are minimum nested 2-covers, then $(S \cap S', T + T')$ and $(S + S', T \cap T')$ are also minimum 2-covers and the former is nested. Hence, there exists a unique minimum nested 2-cover (S^*, T^*) such that $S^* \leq S$ and $T \leq T^*$ for any minimum nested 2-cover (S, T) [3, Lemma 4.9]. This nested 2-cover (S^*, T^*) is called the *dominant 2-cover*. The dominant 2-cover plays an important role in the weighted fractional matroid parity algorithm by Gijswijt and Pap [9].

2.2 Non-commutative Rank

Let A be a linear matrix (1) with $A_1, \ldots, A_m \in \mathbb{K}^{n \times n}$. As described in Section 1, the non-commutative rank (nc-rank) of A, denoted as nc-rank A, is equal to $\frac{1}{d} \operatorname{rank} A^{\{d\}}$ for $d \geq n - 1$, where $A^{\{d\}}$ is the dth-order blow-up (2) of A. In general, the rank and nc-rank of a linear matrix A satisfy $\operatorname{rank} A \leq \text{nc-rank } A \leq 2 \operatorname{rank} A$ [7]. Generalizing the König–Egeváry theorem for bipartite matching and Edmonds' matroid intersection theorem, Fortin and Reutenauer [7] presented the following min-max formulation.

Theorem 3 ([7, Theorem 1]) *For an $n \times n$ linear matrix A, it holds*

$$\text{nc-rank } A = \min \left\{ 2n - s - t : P, Q \in \mathrm{GL}_n(\mathbb{K}), PAQ = \begin{bmatrix} * & * \\ O_{s,t} & * \end{bmatrix} \right\}.$$

Hamada and Hirai [10] rephrased Theorem 3 as follows.

Theorem 4 ([10]) *For an $n \times n$ linear matrix A, it holds*

$$\text{nc-rank } A = \min\{2n - \dim X - \dim Y : X, Y \leq \mathbb{K}^n,\ A_i(X, Y) = \{0\} \text{ for } i \in [m]\}, \tag{5}$$

where $A_i(X, Y) \coloneqq \{x^\top A_i y : x \in X, y \in Y\}$.

The dual problem (5) is called the *minimum vanishing subspace problem* (MVSP). It is known that the MVSP is an example of submodular function minimization on the product of the lattice of all vector subspaces of $\mathbb{K}^n$ and its order-reversed lattice. Namely, if (X, Y) and (X', Y') attain the minimum, so do $(X + X', Y \cap Y')$ and $(X \cap X', Y + Y')$. Using this property, we can show the following.

Lemma 5 *For an $n \times n$ skew-symmetric linear matrix A, we have*

$$\text{nc-rank } A = \min \left\{ 2n - s - t : P \in \mathrm{GL}_n(\mathbb{K}), PAP^\top = \begin{array}{c} \\ n-s \\ s-t \\ t \end{array} \begin{array}{c} \begin{array}{ccc} n-s & s-t & t \end{array} \\ \begin{bmatrix} * & * & * \\ * & * & O \\ * & O & O \end{bmatrix} \end{array} \right\}. \tag{6}$$

2.3 Linear Algebra Toolbox

We collect useful tools in linear algebra. First, we deal with the Kronecker product. Recall that the Kronecker product of an $n \times m$ matrix $A = (a_{ij})$ and a $p \times q$ matrix B is an $np \times mq$ matrix

$$A \otimes B \coloneqq \begin{bmatrix} a_{11}B & \cdots & a_{1m}B \\ \vdots & \ddots & \vdots \\ a_{n1}B & \cdots & a_{nm}B \end{bmatrix}.$$

For matrices A, B, C, D, E, and F of such size that ABC and DEF are defined, the Kronecker product satisfies the *mixed-product property* $(ABC) \otimes (DEF) = (A \otimes D)(B \otimes E)(C \otimes F)$.

Next, let $A = (a_{ij})$ be an $n \times n$ skew-symmetric matrix with n being even. The *Pfaffian* of A is

$$\operatorname{pf} A \coloneqq \sum_{\sigma \in F_n} \operatorname{sgn} \sigma \prod_{i \in [n]:even} a_{\sigma(i-1)\sigma(i)},$$

where F_n is the set of permutations $\sigma : [n] \to [n]$ such that $\sigma(1) < \sigma(3) < \cdots < \sigma(n)$ and $\sigma(i-1) < \sigma(i)$ for even $i \in [n]$. Let $\operatorname{pf} A = 0$ when n is odd. The Pfaffian satisfies $(\operatorname{pf} A)^2 = \det A$, meaning that A is nonsingular if and only if $\operatorname{pf} A \neq 0$, and the following generalization of the Cauchy–Binet formula.

Proposition 6 ([13]) *For skew-symmetric $A \in K^{n \times n}$ and $B \in K^{m \times n}$, it holds*

$$\operatorname{pf} BAB^\top = \sum_J \det B[*, J] \operatorname{pf} A[J],$$

where J runs over all row (column) subsets of A of size m.

3 Non-commutative Rank and Fractional Linear Matroid Parity

In this section, we show the following non-commutative and fractional counterpart to Theorem 1.

Theorem 7 *Let P be a fractional matroid parity polytope and A the corresponding matrix representation* (3)*. Then, we have*

$$\text{nc-rank}\, A = 2 \max_{y \in P} |y|.$$

Proof: First, we show that nc-rank $A \leq 2 \max_{y \in P} |y|$. Let $B_i = \begin{bmatrix} a_i & b_i \end{bmatrix}$ for $i \in [m]$ and (S, T) be a minimum nested 2-cover. By appropriate change of basis, we can assume that $S = \langle e_1, \ldots, e_s \rangle$ and $T = \langle e_1, \ldots, e_t \rangle$ for $s \leq t$. Since (S, T) is a 2-cover, the column-echelon form of B_i must have one of the following block structures:

$$\begin{matrix} s \\ t-s \\ n-t \end{matrix} \begin{bmatrix} * & * \\ * & \mathbf{0} \\ * & \mathbf{0} \end{bmatrix}, \begin{bmatrix} * & * \\ * & * \\ \mathbf{0} & \mathbf{0} \end{bmatrix}.$$

Note that column operations do not change the wedge product $a_i \wedge b_i$ except for the sign.

We will show that the same change of basis yields a common $(n-t) \times (n-s)$ block for each $a_i \wedge b_i$. If $B_i \sim \begin{bmatrix} * & * \\ * & \mathbf{0} \\ * & \mathbf{0} \end{bmatrix}$, then $a_i \wedge b_i = \begin{matrix} \\ s \\ t-s \\ n-t \end{matrix} \begin{matrix} \begin{matrix} s & t-s & n-t \end{matrix} \\ \begin{bmatrix} * & * & * \\ * & O & O \\ * & O & O \end{bmatrix} \end{matrix}$. If $B_i \sim \begin{bmatrix} * & * \\ * & * \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$, then $a_i \wedge b_i = \begin{matrix} \\ s \\ t-s \\ n-t \end{matrix} \begin{matrix} \begin{matrix} s & t-s & n-t \end{matrix} \\ \begin{bmatrix} * & * & O \\ * & * & O \\ O & O & O \end{bmatrix} \end{matrix}$.
Therefore, the right bottom $(n-t) \times (n-s)$ zero block is common for all $a_i \wedge b_i$. By Theorems 2 and 3, we have nc-rank $A \leq 2n - (n-t) - (n-s) = s + t = 2 \max_{y \in P} |y|$.

We show the other direction. Let $P \in \mathrm{GL}_n(\mathbb{K})$ be an optimal solution in Lemma 5 and s, t $(s \geq t)$ be the values in (6) for P. By appropriate change of basis, we can assume that $P = I_n$. For $p, q \in [n]$ $(p \neq q)$, let us denote by $\det B_i[p, q]$ the 2×2 minor corresponding to the pth and qth rows of B_i. Then, every minor $\det B_i[p, q]$ vanishes for $p > n - s$ and $q > n - t$, because it equals the (p, q)-entry of $a_i \wedge b_i$. This implies that the column-echelon form of B_i must be one of the following block structures:

$$\begin{matrix} n-s \\ s-t \\ t \end{matrix} \begin{bmatrix} * & * \\ * & \mathbf{0} \\ * & \mathbf{0} \end{bmatrix}, \begin{bmatrix} * & * \\ * & * \\ \mathbf{0} & \mathbf{0} \end{bmatrix}.$$

Therefore, letting $S = \langle e_1, \ldots, e_s \rangle$ and $T = \langle e_1, \ldots, e_t \rangle$, we can conform that (S, T) is a nested 2-cover with $\dim S + \dim T = 2n - s - t$. This completes the proof. □

Algorithm 1 Simple algebraic algorithm for the fractional linear matroid parity problem

Input: A fractional matroid parity polytope P given as $a_i, b_i \in \mathbb{K}^n$ $(i \in [m])$ and a finite subset R of $\mathbb{K}$.
Output: The lexicographically minimum maximum fractional matroid matching in P

Let $A = \sum_{i=1}^{m} x_i(a_i \wedge b_i)$.
Estimate $\rho_A(\cdot)$ by substituting elements of R uniformly at random in the following.
$r \coloneqq \rho_A(\mathbf{1})$
$y \leftarrow \mathbf{1}$
for $i = 1, \ldots, m$ **do**
    if $\rho_A\big(y - \frac{1}{2}e_i\big) = r$ **then**
        $y_i \leftarrow \frac{1}{2}$
        if $\rho_A(y - \frac{1}{2}e_i) = r$ **then**
            $y_i \leftarrow 0$
return y

4 Algebraic Algorithm

In this section, we present an algebraic algorithm for the fractional linear matroid parity problem that outputs not only the optimal value but also an optimal solution.

4.1 Algorithm Description

Let P be the fractional matroid parity polytope defind from lines $\ell_i = \langle a_i, b_i \rangle$ $(i \in [m])$ and A be the corresponding matrix representation (3). For a half-integral vector $y \in \big\{0, \frac{1}{2}, 1\big\}^m$, let

$$A^{\{2\}}(y) \coloneqq \sum_{i=1}^{m} Y_i \otimes (a_i \wedge b_i),$$

where $Y_i = U_i U_i^\top$ and U_i is a $2 \times 2y_i$ matrix with indeterminates in its entries for $i \in [m]$. We define $Y_i = O$ if $y_i = 0$. Namely, $A^{\{2\}}(y)$ is the matrix obtained by substituting the 2×2 symmetric matrix Y_i of rank $2y_i$ into X_i in the second-order blow-up $A^{\{2\}}$ for $i \in [m]$. Note that $A^{\{2\}}(y)$ is skew-symmetric. We let $\rho_A(y) \coloneqq \operatorname{rank} A^{\{2\}}(y)$.

Algorithm 1 describes the presented algebraic algorithm. The algorithm iteratively computes $\rho_A(y)$ for different $y \in \big\{0, \frac{1}{2}, 1\big\}^m$, which can be efficiently performed via the random substitution from a finite subset $R \subseteq \mathbb{K}$. We will show that the value $r = \rho_A(\mathbf{1})$ computed in Algorithm 1 is four times the cardinality of maximum fractional matroid matching, and $\rho_A(y)$ computed in Algorithm 1 is equal to r if and only if there exists $z \in \big\{0, \frac{1}{2}, 1\big\}^m$ such that $z \leq y$, $z \in P$, and $|z| = \frac{r}{4}$. Thus, Algorithm 1 finds a maximum fractional matroid matching via the search-to-decision reduction. Specifically, the algorithm outputs the *lexicographically minimum* optimal solution. In the rest of this section, we give proofs of these facts and then show the following conclusion.

Theorem 8 *If $|R| \geq 16mn$, Algorithm 1 finds the lexicographically minimum vector among all maximum fractional matroid matchings in P in $O(n^\omega + mn^2)$ time with probability at least $\frac{1}{2}$.*

4.2 Characterizing Rank of Second-order Blow-up

By the skew-symmetricity, $\operatorname{rank} A^{\{2\}}(y)$ is equal to the maximum size of a nonsingular principal submatrix. We first give an expansion formula of the Pfaffian of nonsingular principal submatrices of $A^{\{2\}}(y)$ and use it to characterize the rank of $A^{\{2\}}(y)$. Let $B_i = \begin{bmatrix} a_i & b_i \end{bmatrix}$ for $i \in [m]$.

Lemma 9 *For $y \in \left\{0, \frac{1}{2}, 1\right\}^m$ and $I \subseteq [2n]$, it holds*

$$\operatorname{pf} A^{\{2\}}(y)[I] = \sum_{\substack{z \in \left\{0,\frac{1}{2},1\right\}^m : (J_1,\ldots,J_m) \in \mathcal{J}^y(z) \\ |z| = \frac{|I|}{4},\, z \le y}} \sum \tau_{J_1,\ldots,J_m}, \tag{7}$$

where $\mathcal{J}^y(z)$ is the family of m-tuples $(J_1, \ldots, J_m)$ such that

$$J_i = \begin{cases} \{1,2,3,4\} & (z_i = 1), \\ \{1,2\} \text{ or } \{3,4\} & \left(y_i = 1, z_i = \frac{1}{2}\right), \\ \{1,2\} & \left(y_i = z_i = \frac{1}{2}\right), \\ \emptyset & (z_i = 0), \end{cases}$$

$$\tau_{J_1,\ldots,J_m} = \det\left[(U_1 \otimes B_1)[I, J_1] \cdot \cdots \cdot (U_m \otimes B_m)[I, J_m]\right], \tag{8}$$

and U_i is the $2 \times 2y_i$ matrix given in Section 4.1.

Proof: The wedge product $a_i \wedge b_i$ is written as $B_i \Delta B_i^\top$ with $\Delta = \begin{bmatrix} 0 & +1 \\ -1 & 0 \end{bmatrix}$. Using the mixed-product property, we obtain $Y_i \otimes (a_i \wedge b_i) = \left(U_i I_{2y_i} {U_i}^\top\right) \otimes \left(B_i \Delta B_i^\top\right) = (U_i \otimes B_i)(I_{2y_i} \otimes \Delta)(U_i \otimes B_i)^\top$ and $A^{\{2\}}(y) = B(y)D(y)B(y)^\top$, where

$$B(y) = \left[U_1 \otimes B_1 \cdot \cdots \cdot U_m \otimes B_m\right], \quad D(y) = \begin{bmatrix} I_{2y_1} \otimes \Delta & & \\ & \ddots & \\ & & I_{2y_m} \otimes \Delta \end{bmatrix}. \tag{9}$$

Thus, we have $A^{\{2\}}(y)[I] = B(y)[I, *]D(y)B(y)[I, *]^\top$. Applying Proposition 6, we obtain

$$\operatorname{pf} A^{\{2\}}(y)[I] = \operatorname{pf} B(y)[I, *]D(y)B(y)[I, *]^\top = \sum_J \det B(y)[I, J] \operatorname{pf} D(y)[J], \tag{10}$$

where J runs over all column subsets in $B(y)$ of cardinality $|I|$. Letting J_i be the columns of $(U_i \otimes B_i)[I, *]$ in $B(y)[I, J]$, we have

$$\operatorname{pf} A^{\{2\}}(y)[I] = \sum_{\substack{(J_1,\ldots,J_m): \\ \sum_{i=1}^m |J_i| = |I|}} \tau_{J_1,\ldots,J_m} \prod_{i=1}^m \operatorname{pf}(I_{2y_i} \otimes \Delta)[J_i].$$

Let $z_i = \frac{|J_i|}{4}$ for $i \in [m]$. Now $\operatorname{pf}(I_{2y_i} \otimes \Delta)[J_i] \in \{0, 1\}$ and it does not vanish if and only if $|J_i| \le 4y_i$ and $J_i = \{1,2,3,4\}$, $\{1,2\}$, $\{3,4\}$ (allowed only when $y_i = 1$), or $\emptyset$. Thus, z corresponding to non-vanishing terms is a half-integral vector with $z \le y$. The equation (10) is represented as (7) in this way. □

Corollary 10 *For $y \in \left\{0, \frac{1}{2}, 1\right\}^m$, $\rho_A(y)$ is equal to four times the maximum cardinality of $z \in \left\{0, \frac{1}{2}, 1\right\}^m$ such that $z \le y$ and $B(z)$ is of column-full rank, where $B(z)$ is defined by* (9).

Proof: Since $A^{\{2\}}(y)$ is skew-symmetric, $\rho_A(y) = \operatorname{rank} A^{\{2\}}(y)$ is equal to the maximum cardinality of $I \subseteq [2n]$ such that $A^{\{2\}}(Y)[I]$ is nonsingular. Fix $I \subseteq [2n]$. By Lemma 9, $\operatorname{pf} A^{\{2\}}(y)[I]$ is expanded as (7). Now the summand (8) of (7) for z is the same polynomial as $\det B(z)[I]$ up to the indeterminates' labeling. Thus, if $B(z)[I]$ is singular for any $z \in \left\{0, \frac{1}{2}, 1\right\}^m$ with $z \le y$ and $|z| = \frac{|I|}{4}$, the principal submatrix $A^{\{2\}}(y)[I]$ must be singular. Conversely, if there exists $z \in \left\{0, \frac{1}{2}, 1\right\}^m$ such that $z \le y$, $|z| = \frac{|I|}{4}$, and $B(z)[I]$ is nonsingular, the principal submatrix $A^{\{2\}}(y)[I]$ becomes nonsingular because different z and $(J_1, \ldots, J_m) \in \mathcal{J}^y(z)$ yield summands (8) that do not cancel out.

To summarize, $A^{\{2\}}(y)[I]$ is nonsingular if and only if there exists $z \in \left\{0, \frac{1}{2}, 1\right\}^m$ with $z \le y$ such that $B(z)[I]$ is nonsingular. Finding a maximum I satisfying these conditions, we obtain the claim. □

4.3 Full-column Rankness and Half-integral Fractional Matroid Matchings

Corollary 10 characterizes the $\rho_A(y)$ value in terms of the column full rankness of $B(y)$. We next relate $B(y)$ and half-integral points in P.

Lemma 11 *A half-integral vector $y \in \{0, \frac{1}{2}, 1\}^m$ is in P if $B(y)$ is of full-column rank.*

Proof: We check $\sum_{i=1}^m \dim(S \cap \ell_i) y_i \leq \dim S$ for $S \leq \mathbb{K}^n$. Recall that $B(y)$ is constructed from $U = (U_1, \ldots, U_m)$ where each U_i is of size $2 \times 2y_i$ with indeterminates. Let $\mathbb{F} = \mathbb{K}(U)$ be the rational function field obtained by adjoining the entries of U to $\mathbb{K}$, and $\hat{S} = \mathbb{F} \otimes_{\mathbb{K}} S$ and $\hat{\ell}_i = \mathbb{F} \otimes_{\mathbb{K}} \ell_i$ be the scalar extensions of S and ℓ_i, respectively. Then, $\dim S = \dim \hat{S}$ holds and

$$\dim(S \cap \ell_i) = \dim \mathbb{F} \otimes_{\mathbb{K}} (S \cap \ell_i) = \dim\bigl(\hat{S} \cap \hat{\ell}_i\bigr). \tag{11}$$

Let $W_i = (\mathbb{F}^2 \otimes_{\mathbb{F}} \hat{S}) \cap \mathrm{Im}_{\mathbb{F}}(U_i \otimes B_i)$, where $\mathbb{F}^2 \otimes_{\mathbb{F}} \hat{S}$ is the tensor product of $\mathbb{F}^2$ and $\hat{S}$ and $\mathrm{Im}_{\mathbb{F}}(U_i \otimes B_i)$ is the image space of the matrix $U_i \otimes B_i$ over $\mathbb{F}$. Using the distributive inequality $V_1 \cap (V_2 + V_3) \supseteq (V_1 \cap V_2) + (V_1 \cap V_3)$ of vector spaces V_1, V_2, and V_3, we have

$$\mathbb{F}^2 \otimes_{\mathbb{F}} \hat{S} \supseteq \bigl(\mathbb{F}^2 \otimes_{\mathbb{F}} \hat{S}\bigr) \cap \mathrm{Im}\, B(y) = \bigl(\mathbb{F}^2 \otimes_{\mathbb{F}} \hat{S}\bigr) \cap \sum_{i=1}^m \mathrm{Im}(U_i \otimes B_i) \supseteq \sum_{i=1}^m W_i.$$

This means that

$$2 \dim S = 2 \dim \hat{S} = \dim\bigl(\mathbb{F}^2 \otimes_{\mathbb{F}} \hat{S}\bigr) \geq \dim \sum_{i=1}^m W_i = \sum_{i=1}^m \dim W_i, \tag{12}$$

where the last equality is due to the assumption that $B(y)$ is of full-column rank.

We next show $\dim W_i = 2 \dim_{\mathbb{F}}(\hat{S} \cap \hat{\ell}_i) y_i$ for $i \in [m]$, which completes the proof by (11) and (12). Using $\mathrm{Im}_{\mathbb{F}} B_i = \hat{\ell}_i$, we obtain $\mathrm{Im}_{\mathbb{F}}(U_i \otimes B_i) = \mathrm{Im}_{\mathbb{F}} U_i \otimes_{\mathbb{F}} \mathrm{Im}_{\mathbb{F}} B_i = \mathrm{Im}_{\mathbb{F}} U_i \otimes_{\mathbb{F}} \hat{\ell}_i$. Thus, we have

$$W_i = \bigl(\mathbb{F}^2 \otimes_{\mathbb{F}} \hat{S}\bigr) \cap \bigl(\mathrm{Im}_{\mathbb{F}} U_i \otimes_{\mathbb{F}} \hat{\ell}_i\bigr) = \bigl(\mathbb{F}^2 \cap \mathrm{Im}_{\mathbb{F}} U_i\bigr) \otimes_{\mathbb{F}} \bigl(\hat{S} \cap \hat{\ell}_i\bigr) = \mathrm{Im}_{\mathbb{F}} U_i \otimes_{\mathbb{F}} \bigl(\hat{S} \cap \hat{\ell}_i\bigr)$$

and hence $\dim W_i = (\dim \mathrm{Im}_{\mathbb{F}} U_i) \cdot \bigl(\dim\bigl(\hat{S} \cap \hat{\ell}_i\bigr)\bigr) = 2y_i \cdot \dim\bigl(\hat{S} \cap \hat{\ell}_i\bigr)$ holds as required. □

The converse of Lemma 11 holds for *extreme* points in P. The extremality is needed to use the characterization of extreme fractional matroid matchings given by Chang et al. [3]. We omit the proof due to space limitation; see the full paper for complete proof.

Lemma 12 *The matrix $B(y)$ with $y \in \{0, \frac{1}{2}, 1\}^m$ is of full-column rank if y is an extreme point of P.*

It is unknown whether $B(y)$ is of full-column rank even for a half-integral but non-extreme $y \in P$. Nevertheless, our weak characterization is enough to show the validity of Algorithm 1.

4.4 Proof of Theorem 8

Corollary 10 and Lemmas 11 and 12 are aggregated into the following lemma.

Lemma 13 *For $y \in \{0, \frac{1}{2}, 1\}^m$, $\rho_A(y) \leq 4 \max\{|z| : z \in P, z \leq y\}$ holds. The equality is attained if there exists extreme $z \in P$ with $z \leq y$ that attains the maximum in $\max\{|z| : z \in P, z \leq y\}$.*

Proof: By Corollary 10, $\rho_A(y)$ is equal to four times the maximum cardinality of $z \in \{0, \frac{1}{2}, 1\}^m$ such that $z \leq y$ and $B(z)$ is of column-full rank. Since such z is in P by Lemma 11, we have the inequality. Next, suppose that there exists extreme $z^* \in P$ with $z^* \leq y$ that attains the maximum in $\max\{|z| : z \in P, z \leq y\}$. By Lemma 12, $B(z^*)$ is of full column rank, meaning $4|z^*| \leq \rho_A(y)$ by

Corollary 10. Now we have $\rho_A(y) \le 4\max\{|z| : z \in P, z \le y\} = 4|z^*| \le \rho_A(y)$ and hence the equality is attained. □

Now we are ready to prove Theorem 8.

PROOF: First, we show the validity of the algorithm assuming that the exact value of ρ_A can be computed. Let y^* be the lexicographically minimum point among all points in P with maximum cardinality. Note that y^* is half-integral since y^* is extreme. Let $y^{(0)} = \mathbf{1}$ and $y^{(i)}$ denote y in Algorithm 1 at the end of the ith iteration for $i \in [m]$. We show by induction on i the following claim: for every $i = 0, \ldots, m$, it holds $y^{(i)}_j = y^*_j$ if $j \le i$ and $y^{(i)}_j = 1$ if $j > i$. Then Theorem 8 is obtained as the case for $i = m$.

The claim for $i = 0$ is trivial. Suppose that the claim is true for $i - 1$ and consider the case for i. Let y be the candidate solution given to ρ_A in Line 6 of Algorithm 1. Suppose the case when $y^*_i = 1$. Then y is lexicographically smaller than y^* by the inductive assumption. This means that that there is no maximum point $z \in P$ satisfying $z \le y$. By Lemma 13, we have

$$\rho_A(y) \le 4\max\{|z| : z \in P, z \le y\} < 4\max\{|z| : z \in P\} = \rho_A(\mathbf{1})$$

and thus the ith iteration in Algorithm 1 does not execute Lines 7–9 and $y^{(i)}_i$ is fixed to 1. Suppose the case when $y^*_i \le \frac{1}{2}$. Then $y^* \le y$ by the inductive assumption. We have

$$\max\{|z| : z \in P\} = |y^*| \le \max\{|z| : z \in P, z \le y\} \le \max\{|z| : z \in P\},$$

which means that y^* attains the maximum in $\max\{|z| : z \in P, z \le y\}$. Since y^* is extreme, we have

$$\rho_A(y) = 4\max\{|z| : z \in P, z \le y\} = 4|y^*| = \rho_A(\mathbf{1})$$

by Lemma 13. Thus the ith iteration goes to Line 7. The same argument can be applied to the conditional branch in Line 8. This completes the proof of validity assuming that all the evaluations of ρ_A is exact.

Next, we analyze the probability of success when we estimate $\rho_A(y)$ by substituting uniform random elements from R to the entries of U_i. Since each entry of $A^{\{2\}}(y) = \sum_{i=1}^m U_i U_i^\top \otimes (a_i \wedge b_i)$ is quadratic, the degree of any non-vanishing $k \times k$ minor is $2k$ for any $k \le 2n$. Therefore, the probability that such a non-vanishing minor remains non-vanishing after the random substitution from R is at least $1 - \frac{4n}{|R|}$ by the Schwartz-Zippel lemma. Since there are at most $2m$ evaluations of ρ_A, it suffices to take $|R| = 16mn$ to guarantee that all the evaluations of ρ_A during the algorithm are correct with probability at least $\frac{1}{2}$. Thus, the output of the algorithm is correct with probability at least $\frac{1}{2}$.

Finally, we analyze the time complexity. The value of ρ_A can be evaluated in $O(n^2)$ time as follows. Let y be a tentative solution, $y' = y - \frac{1}{2}e_i$ be a candidate solution in the algorithm, and U_i be a random matrix of size $2 \times 2y_i$ for $i \in [m]$. Suppose that we already have an LU decomposition LU of $A^{\{2\}}(y)$. Then, to evaluate $\rho_A(y')$, it suffices to compute an LU decomposition of $M' = LU + D_i \otimes (a_i \wedge b_i)$, where

$$D_i = \begin{cases} -U_i[*, \{1\}]U_i[*, \{1\}]^\top & \big(y_i = 1, y'_i = \frac{1}{2}\big), \\ -U_i U_i^\top & \big(y_i = \frac{1}{2}, y'_i = 0\big). \end{cases}$$

Note that the rank of D_i is 1, so that of $D_i \otimes (a_i \wedge b_i)$ is 2. Hence, we can use an update formula [1] for an LU decomposition to compute $\operatorname{rank} M'$ in $O(n^2)$ time. For the first evaluation of ρ_A, we simply compute an LU decomposition in $O(n^\omega)$ time. Thus, the time complexity of Algorithm 1 is $O(n^\omega + mn^2)$. □

References

[1] J. M. Bennett. "Triangular factors of modified matrices". In: *Numerische Mathematik* 7.3 (1965), pp. 217–221.

[2] S. Y. Chang, D. C. Llewellyn, and J. H. Vande Vate. "Matching 2-lattice polyhedra: finding a maximum vector". In: *Discrete Mathematics* 237.1-3 (2001), pp. 29–61.

[3] S. Y. Chang, D. C. Llewellyn, and J. H. Vande Vate. “Two-lattice polyhedra: duality and extreme points”. In: *Discrete Mathematics* 237.1-3 (2001), pp. 63–95.

[4] H. Y. Cheung, L. C. Lau, and K. M. Leung. “Algebraic algorithms for linear matroid parity problems”. In: *ACM Transactions on Algorithms* 10.3 (2014), pp. 1–26.

[5] H. Derksen and V. Makam. “Polynomial degree bounds for matrix semi-invariants”. In: *Advances in Mathematics* 310 (2017), pp. 44–63.

[6] J. Edmonds. “Systems of distinct representatives and linear algebra”. In: *Journal of research of the National Bureau of Standards* 71B.4 (1967), pp. 241–245.

[7] M. Fortin and C. Reutenauer. “Commutative/noncommutative rank of linear matrices and subspaces of matrices of low rank”. In: *Séminaire Lotharingien de Combinatoire* 52 (2004), B52f.

[8] A. Garg, L. Gurvits, R. Oliveira, and A. Wigderson. “Operator scaling: theory and applications”. In: *Foundations of Computational Mathematics* 20.2 (2020), pp. 223–290.

[9] D. Gijswijt and G. Pap. “An algorithm for weighted fractional matroid matching”. In: *Journal of Combinatorial Theory, Series B* 103.4 (2013), pp. 509–520.

[10] M. Hamada and H. Hirai. “Computing the nc-rank via discrete convex optimization on CAT(0) spaces”. In: *SIAM Journal on Applied Algebra and Geometry* 5.3 (2021), pp. 455–478.

[11] N. J. A. Harvey. “Algebraic algorithms for matching and matroid problems”. In: *SIAM Journal on Computing* 39.2 (2009), pp. 679–702.

[12] P. Hrubeš and A. Wigderson. “Non-commutative arithmetic circuits with division”. In: *Theory of Computing* 11.14 (2015), pp. 357–393.

[13] M. Ishikawa and M. Wakayama. “Minor summation formula of Pfaffians”. In: *Linear and Multilinear Algebra* 39.3 (1995), pp. 285–305.

[14] G. Ivanyos, Y. Qiao, and K. V. Subrahmanyam. “Constructive non-commutative rank computation is in deterministic polynomial time”. In: *Computational Complexity* 27.4 (2018), pp. 561–593.

[15] V. Kabanets and R. Impagliazzo. “Derandomizing polynomial identity tests means proving circuit lower bounds”. In: *Computational Complexity* 13.1–2 (2004), pp. 1–46.

[16] E. L. Lawler. *Combinatorial Optimization: Networks and Matroids*. New York: Holt, Rinehart and Winston, 1976.

[17] L. Lovász. “On determinants, matchings, and random algorithms”. In: *Fundamentals of Computation Theory*. Ed. by L. Budach. Berlin: Akademie-Verlag, 1979.

[18] L. Lovász. “Singular spaces of matrices and their application in combinatorics”. In: *Bulletin of the Brazilian Mathematical Society. New Series. Boletim da Sociedade Brasileira de Matematica* 20.1 (1989), pp. 87–99.

[19] J. T. Schwartz. “Fast probabilistic algorithms for verification of polynomial identities”. In: *Journal of the ACM* 27.4 (1980), pp. 701–717.

[20] N. Tomizawa and M. Iri. “An algorithm for determining the rank of a triple matrix product AXB with application to the problem of discerning the existence of the unique solution in a network (in Japanese)”. In: *Electronics and Communications in Japan* 57.11 (1974), pp. 50–57.

[21] W. T. Tutte. “The factorization of linear graphs”. In: *Journal of the London Mathematical Society. Second Series* s1-22.2 (1947), pp. 107–111.

[22] J. H. Vande Vate. “Fractional matroid matchings”. In: *Journal of Combinatorial Theory, Series B* 55.1 (1992), pp. 133–145.

[23] R. Zippel. “Probabilistic algorithms for sparse polynomials”. In: *Symbolic and Algebraic Computation*. Ed. by E. W. Ng. Vol. 72. Lecture Notes in Computer Science. Berlin: Springer, 1979, pp. 216–226.

Common systems of two equations over the binary field

Daniel Král'[1]

Faculty of Informatics
Masaryk University
Botanická 68A, 602 00 Brno, Czech Republic
dkral@fi.muni.cz

Ander Lamaison[1]

Faculty of Informatics
Masaryk University
Botanická 68A, 602 00 Brno, Czech Republic
lamaison@fi.muni.cz

Péter Pál Pach[2]

Department of Computer Science and
Information Theory
Budapest University of Technology and
Economics
Műegyetem rkp. 3., H-1111 Budapest, Hungary;
MTA-BME Lendület Arithmetic Combinatorics
Research Group, ELKH
Műegyetem rkp. 3., H-1111 Budapest, Hungary;
and
Extremal Combinatorics and Probability Group
(ECOPRO), Institute for Basic Science (IBS)
Daejeon, South Korea
ppp@cs.bme.hu

Abstract: A system of linear equations over a finite field $\mathbb{F}_q$ is said to be common if, among all two-colorings of $\mathbb{F}_q^n$, the uniform random coloring minimizes the number of monochromatic solutions asymptotically. The notion of common systems of linear equations was introduced by Saad and Wolf, as an analogue to the well-studied notion of common graphs.

Fox, Pham and Zhao characterized the common systems consisting of one equation. We study systems consisting of two equations over the binary field $\mathbb{F}_2$. We characterize, up to a finite number of cases, which systems with an odd number of variables are common. Our characterization answers a question by Kamčev, Liebenau and Morrison in the affirmative way whether there exist common systems of equations that are not translation invariant.

Keywords: common systems, monochromatic solutions, Sidorenko's conjecture

[1]The work of D.K. and A.L. has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 648509). This publication reflects only its authors' view; the European Research Council Executive Agency is not responsible for any use that may be made of the information it contains. The authors were also supported by the MUNI Award in Science and Humanities of the Grant Agency of Masaryk University.

[2]P.P.P. was supported by the Lendület program of the Hungarian Academy of Sciences (MTA), the National Research, Development and Innovation Office NKFIH (Grant Nr. K124171 and K129335) and by the Institute for Basic Science (Grant Nr. IBS-R029-C4).

New results on synchronized TSP

Gyula Pap[1]

Department of Operations Research
Eötvös University, Budapest, Hungary
gyula.pap@ttk.elte.hu

Abstract: In synchronized TSP we consider multiple agents walking around in a graph without bumping into each other, a notion introduced in [2]. We can consider optimization problems relating to synchronized TSP, and some lower and upper bounds have been proved in [2]. In this paper we extend on these results in two different ways. First we consider so-called periodic agencies, and prove that an optimal periodic agency can be found for bounded treewidth graphs and bounded periodicity. Second we elaborate on the maximum number of agents problem from [2], and prove that for connected graphs with minimum degree 3 there is an agency of at least $n - 5$.

Keywords: traveling salesman, optimization, spanning tree

1 Introduction

The notion of the synchronized traveling salesman problem was introduced in [2], in which there is a set of so-called agents walking around in an undirected graph, and avoiding a collision with each other, but each of them performing a tour of all vertices of the graph. In many cases considered, a feasible agency is constructed in a way that the trajectory of the agents is repeated after a delay, and thus resulting in this bunch of trajectories avoiding a collision. An agency that is constructed in this way is called an *periodic agency* and in this paper we investigate the problem of finding an optimal periodic agency in the input graph.

2 Problem setting

In the notation and terminology we go by those introduced in [2], which is summarized as follows. Let $G = (V, E)$ be an undirected graph with $n = |V|$. A sequence $v(0), v(1), v(2), \ldots, v(T)$ of nodes $v(t) \in V$ is called a walk (with parking) if for all $t = 0, 1, \ldots, T-1$ we have either $v(t) = v(t+1)$ or $v(t)v(t+1) \in E$. A sequence $v(0), v(1), v(2), \ldots, v(T)$ of nodes $v(t) \in V$ is called a walk (without parking) if for all $t = 0, 1, \ldots, T-1$ we have $v(t)v(t+1) \in E$. For a walk, T is called the time horizon.

A sequence $v(0), v(1), v(2), \ldots, v(T)$ of nodes $v(t) \in V$ is called a traveling salesman tour (with/without parking), if it is a walk (with/without parking) in the graph such that every node appears at least once, and the tour returns to its initial node, that is, $v(0) = v(T)$. A traveling salesman tour with parking is called a tour, for short. As a tour returns to its initial node, we may consider a tour by time units modulo T (which is similar to picturing a tour as if it were to repeat all over after the time horizon).

Of course in the usual setting of the traveling salesman problem, there is no need for parking, because it is just a waste of time or cost; here in our setting, however, parking may be needed to avoid two salesmen of crashing into each other: one of them would wait until the other one passes a node or an

[1]Member of MTA-ELTE Egerváry Research Group, Department of Operations Research, Eötvös University, Pázmány Péter sétány 1/C, Budapest, Hungary, H-1117. E-mail: gyula.pap@ttk.elte.hu. This research is supported by the Hungarian National Research, Development and Innovation Office grant NKFI-132524.

edge, and move on afterwards. Vaguely speaking, the point is that we introduce a setting in which there are multiple salesmen touring the same graph at the same time so that they are not allowed to crash into each other. In this setting it makes a lot of sense to allow parking, and this is what we do in this paper.

In the synchronized traveling salesman problem, we consider an **"agency"** of a number of salesmen each one of which has to do a tour with the same time horizon, though they need to start from different initial nodes, and must not "crash" into each other. Essentially there is a unit capacity for each node or each edge. More precisely, we define an agency as follows.

Let $k, T \in \mathbb{Z}_+$ be positive integers, where k denotes the number of salesmen, or agents, and T denotes the joint time horizon. Let $a_i(t) \in V$ be the node where agent i is supposed to be at time t, where $i = 1, 2, \ldots, k$, and $t = 0, 1, \ldots, T$. This triple k, T, a_i is called an agency with time horizon T and k agents if for any fixed i, $a_i(0), a_i(1), \ldots, a_i(T)$ gives a tour (with parking). In practical terms, each i denotes an agent that moves along the unit-length edges of the graph, so that every agent makes a traveling salesman tour of time horizon T.

If i and j are agents from the same agency, $i \neq j$, then we say that these agents i and j crash in a node v at time t if $v = a_i(t) = a_j(t)$. If i and j are agents from the same agency, $i \neq j$, then we say that these agents i and j crash in an edge $uv \in E$ at time t if $v = a_i(t) = a_j(t+1), u = a_i(t+1) = a_j(t)$ or $v = a_i(t+1) = a_j(t), u = a_i(t) = a_j(t+1)$.

An agency is called a *feasible agency* if there is no crash between any pair of agents in neither an edge nor a node. In practical terms, this may be understood as a set of agents moving along the unit-length edges of the graph so that they avoid crashing into each other, but each of them manages to visit every node at least once, before finally arriving at their respective nodes of origin.

3 Maximum number of agents

In [2] we determined that if the input graph G is a tree, then the maximum number of agents in a feasible agency can be determined by the maximum stretch in the tree (here there is no bound on the time horizon). Here we use an equivalent definition: for a tree T let $s(T)$ denote the maximum of $|V(P)|$ where P is a subpath of T such that all nodes of P have degree at least 2, and all internal nodes of P have degree equal to 2. This value of $s(T)$ defines a kind of "stretch" of the tree: the longest path in the tree that does not go through any branching nodes and excludes leaf nodes. In [2] we proved the following result.

Theorem 1 (Pap, Varnyú [2]) *Given any tree T on at least 3 vertices, the maximum number of agents in a tree T is equal to $|V(T)| - s(T) - 1$.*

We would, however, like to determine the maximum number of agents for any graph, not just trees. One possibility is that we determine a spanning tree T with $s(T)$ as small as possible, and then use the above result to construct an agency with $|V(T)| - s(T) - 1$ agents. This provides a lower bound on the maximum number of agents, but the question is, how good a lower bound this is?

$s(T) = 1$ if and only if T is a star. If $s(T) \leq 2$ then there are no two consecutive nodes of degree 2. More generally, the maximum number of degree 2 nodes of T forming a connected path is equal to $s(T) - 1$ or $s(T) - 2$. So our problem is closely related with finding a spanning tree that minimizes the number of degree 2 nodes forming a path.

Following the terminology of Lyngsie, Merker [1], a tree is called a homeomorphically irreducible, or HIT, if it has no nodes of degree equal to 2. Thus for a every HIT T, we get that $s(T) \leq 2$. As an approach of constructing a feasible agency maximizing the number of agents, we may try to find a HIT spanning tree. However, we bump into the difficulty that this problem is NP-hard:

Claim 2 *The problem of finding a HIT spanning tree is NP-complete, even for planar graphs.*

Proof: We can prove this by a reduction from the Hamiltonian path problem. Suppose we would want to find a Hamiltonian path in an input graph $G = (V, E)$ from s to t for some $s, t \in V$. Then we define

an auxiliary graph as follows: $V' := V \cup \{v' : v \in V\} \cup \{s'', t''\}$ and $E' := E \cup \{vv' : v \in V\} \cup \{ss'', tt''\}$. It is pretty easy to see that $G' := (V', E')$ contains a HIT spanning tree if and only if the original graph G contains an s-t Hamiltonian path. □

To provide a positive result, Lyngsie, Merker [1] considered the problem of finding spanning trees in a graph so that the tree has as few consecutive adjacent nodes of degree 2 as possible. Clearly this problem is related with our problem above, although there is a difference of 1 or 2 between $s(T)$, and this value. They proved the following result:

Theorem 3 (Lyngsie, Merker [1]) *If a simple connected graph G has minimum degree 3, then it has a spanning tree T such that no three adjacent nodes in T have degree equal to 2 in T.*

Note that if a tree T nos no three adjacent nodes of degree equal to 2, then $s(T) \leq 4$. Together with Theorem 1 we get the following bound on the maximum number of agents.

Theorem 4 *In a simple connected graph with minimum degree at least 3, the maximum number of agents in a feasible agency is at least $n - 5$.*

Thus we have established a lower and upper bounds for a number of problems above, and because these problems turn out to be NP-hard, we only got to find an approximation, but not an exact solution. It is worth noting that these problems are polynomial-time solvable for the special case of bounded treewidth graphs:

Claim 5 *The following problems are fixed parameter tractable for bounded treewidth graphs:*

1. *Find a HIT spanning tree, or conclude that there aren't any.*
2. *Find a spanning tree minimizing the maximum number of consecutive nodes of degree equal to 2.*
3. *Minimizing $s(T)$ for a spanning tree T.*

4 Periodic agencies

In many cases, and earlier results, a feasible agency can be constructed so that after just a few steps, the set of agents will occupy the exact same subset of nodes as in the beginning, albeit in a different permutation. In [2] we constructed a periodic agency for the lower bound in the case when graph G is a tree, and a a different construction resulted in a periodic agency in the case when G is a 3-connected 3-regular graph. This hints to that periodic agencies can provide a good solution in many cases. The definition of periodic agencies goes as follows.

Consider a walk $a_1(0), a_1(1), \cdots, a_1(T-1), a_1(T)$ so that it returns to the initial node $a_1(0) = a_1(T)$ after time horizon T. This would be the trajectory of the first agent. We allow parking in this definition, but the definition also works the same when we don't allow parking. Suppose that T is a multiple of the positive integer λ, which will be the length of the period. Thus we have $T = \lambda\mu$, where μ is another positive integer. The motion of the agency repeats with a period of λ, so for certain remainders modulo λ there will be an agent following agent 1 after that delay. We denote these "delays" (remainders modulo λ) by $0 = j_1 < j_2 < \cdots j_{k_0} \leq \lambda - 1$. So agents $1, 2, \cdots, k_0$ are trailing behind agent 1 at a time delay of $j_1, j_2, \cdots, j_{k_0}$. All this is repeated for any delay that is a multiple of λ, so in total there will be $k = k_0\mu$ agents in this agency. So this means that $a_{\alpha k_0+\beta}(j_\beta + \alpha\lambda + t) = a_1(j_1 + t)$ for any t, α, β. A feasible agency that is constructed this way is called a **λ-periodic agency**.

In [2] all feasible agencies constructed for the purpose of a lower bound are actually periodic agencies. It is thus quite natural to try to find an optimal periodic agency in general, or in special cases. Because of the same reduction from the Hamiltonian cycle problem, the problem of finding a periodic agency with a given number of agents and a given periodicity is NP-hard. However for certain special cases we may prove that a periodic agency can be found in polynomial time.

Theorem 6 *Consider fixed positive integer parameters w, λ. Then there is a polynomial time algorithm to find a λ-periodic agency in a graph with tree-width at most w with a maximum number of agents.*

PROOF: In the minimum treewidth representation of our graph $G = (V, E)$, let T denote the tree, and choose a root $r \in V(T)$. We may assume that T has maximum degree 3, and thus any node has at most two children making it a binary tree. For any $t \in V(T)$ let T_t denote the subtree of T below t. For any $t \in V(T)$ let $V_t \subseteq V$ be the corresponding "bag", and let $U_t := \cup_{q \in T_t} V_q$. By definition, the treewidth is $tw(G) = \max |V_t| - 1$, thus the size of V_t is bounded from above by the fixed constant $|V_t| \leq w + 1$. As usual for bounded treewidth graph algorithms, we use a dynamic programming approach in which we solve a large (but polynomial) number of sub-problems, where the sub-problems correspond to subtrees T_t of the representation tree, and in the end we can put together the final solution.

We specify the subproblems as follows. For some $t \in V(T)$, and a subset $Z \subseteq V_t \times \{0, 1, \cdots, \lambda - 1\}$, and a set of disjoint pairs $M \subseteq \binom{Z}{2}$, let $K(t, Z, M)$ denote the maximum number of agents in a sub-agency in subgraph U_t such that Z is the set of time units occupied at the specific node, modulo λ, and M denotes the set of pairs for which the sub-agency performs a sub-walk. The singletons of Z not appearing in any pair in M are time units occupied at that specific node, by a sub-walk corresponding to some pair in M (it does not matter which pair it is). The number of sub-problems define this way is at most $w \cdot 8^{w\lambda} \cdot (\frac{1}{2}w\lambda)!$, which can be determined by $|V(T)| \leq w$, and counting the number of subsets Z, and counting the number of matchings in Z. This is a fixed constant upper bound, assuming that w and λ are fixed, as of the assumptions of the theorem.

The dynamic programming recursion can be formulated as follows. In T any node has at most two children, and actually we may assume that and thus, to determine the value of $K(t, Z, M)$ we need to consider (at most) two children of t, say t', t''. The optimum sub-agency for t, Z, M can then be determined by choosing one optimum sub-agency for t', Z', M', another sub-agency for t'', Z'', M'', and using any of the edges in $E[V_t]$ to connect between any of these sub-paths. Using any edges in $E[V_t]$ also means that we are committing to the remainder modulo λ the particular edge is used, so there are actually $|E[V_t]|\lambda$ number of possibilities for that, and each of thos possibilities is either chosen or not, so that makes it at most $2^{\binom{w}{2}\lambda}$ number of possibilities to consider. There are $2^{w\lambda}$ choices to choose Z', same number of choses for Z''. There are $4^{w\lambda} \cdot (\frac{1}{2}w\lambda)!$ choices to choose M', same number of choses for M''. We need to do this for all $t \in V(T)$, meaning $|V(T)| \leq |V| = n$ number of choices. The total running time for this algorithm becomes $O(n2^{\binom{w}{2}\lambda}4^{w\lambda}16^{w\lambda} \cdot (\frac{1}{2}w\lambda)!^2) = O(n2^{\binom{w}{2}\lambda}64^{w\lambda} \cdot (\frac{1}{2}w\lambda)!^2)$, which is linear in n, although quite a terrible fixed constant when it comes to the two parameters w, λ. This proves the theorem. □

References

[1] KASPER SZABO LYNGSIE AND MARTIN MERKER, Spanning trees without adjacent vertices of degree 2 *Discrete Mathematics* **342** 12 (2019)

[2] GYULA PAP AND JÓZSEF VARNYÚ, Synchronized Traveling Salesman Problem *J. Graph Algorithms Appl.* **25** 1 (2021)

[3] GYULA PAP , Synchronized Traveling Salesman Problem *IN: Proceedings of The 11th Hungarian-Japanese Symposium on Discrete Mathematics and Its Applications* (2019)

Connected Turán number of trees

Yair Caro

Department of Mathematics
University of Haifa-Oranim, Israel
yacaro@kvgeva.org.il

Balázs Patkós[1]

Alfréd Rényi Institute of Mathematics
Budapest, Hungary
patkos@renyi.hu

Zsolt Tuza [2]

Alfréd Rényi Institute of Mathematics and
University of Pannonia
tuza@dcs.uni-pannon.hu

Abstract: As a variant of the much studied Turán number, $\mathrm{ex}(n, F)$, the largest number of edges that an n-vertex F-free graph may contain, we introduce the connected Turán number $\mathrm{ex}_c(n, F)$, the largest number of edges that an n-vertex connected F-free graph may contain. We focus on the case where the forbidden graph is a tree. The celebrated conjecture of Erdős and Sós states that for any tree T, we have $\mathrm{ex}(n, T) \leq (|T|-2)\frac{n}{2}$. We address the problem how much smaller $\mathrm{ex}_c(n, T)$ can be, what is the smallest possible ratio of $\mathrm{ex}_c(n, T)$ and $(|T|-2)\frac{n}{2}$ as $|T|$ grows. We also determine the exact value of $\mathrm{ex}_c(n, T)$ for small trees, in particular for all trees with at most six vertices. We introduce general constructions of connected T-free graphs based on graph parameters as longest path, matching number, branching number, etc.

Keywords: extremal graph theory, connected graphs, trees

1 Introduction

One of the most studied problems in extremal graph theory is to determine the Turán number $\mathrm{ex}(n, F)$, the largest number of edges that an n-vertex graph can have without containing a subgraph isomorphic to F. In this paper, we study a variant of this parameter: the connected Turán number $\mathrm{ex}_c(n, F)$ is the largest number of edges that a *connected* n-vertex graph can have without containing F as a subgraph. Observe that if F is 2-edge-connected, then any maximal F-free graph G is connected, as if G had at least two components, then adding an edge between them would not create any copy of F. Also, if the chromatic number of F is at least 3, then by the famous theorem by Erdős, Stone, and Simonovits [5, 6], we know that $\mathrm{ex}(n, F)$ is attained asymptotically (and for some graphs precisely) at the Turán graph that is connected. These two observations imply the following proposition.

Proposition 1

1. *If all components of F are 2-edge-connected, then* $\mathrm{ex}(n, F) = \mathrm{ex}_c(n, F)$.
2. *If* $\chi(F) \geq 3$, *then* $\mathrm{ex}_c(n, F) = (1 + o(1))\,\mathrm{ex}(n, F)$.

The asymptotics of $\mathrm{ex}(n, F)$ is unknown for most biparite F (for a general overview of the so-called degenrate Turán problems, see the survey by Füredi and Simonovits [7]). And we do not know the relationship of $\mathrm{ex}(n, F)$ and $\mathrm{ex}_c(n, F)$ for most bipartite F that are not 2-edge-connected. There is a

[1]Research is supported by NKFIH grants SNN 129364 and FK 132060
[2]Research is supported by NKFIH grants SNN 129364

relatively large literature on the Turán number of forests (see e.g. [3, 9, 10, 12, 13]), and in many cases the extremal graphs turned out to be connected, so for those forests F, we have $\mathrm{ex}(n,F) = \mathrm{ex}_c(n,F)$. A wide and important class of connected non-2-edge-connected graphs is the set of trees. A famous conjecture of Erdős and Sós (that appeared in print first in [4]) states that any n-vertex graph with more than $\frac{(k-2)n}{2}$ edges contains any tree T on k vertices. A proof was announced in the early 1990's by Ajtai, Komlós, Simonovits, and Szemerédi, but only arguments of special cases have appeared. A recent survey of these and other degree conditions that imply embeddings of trees is [11]. The universal construction that shows the tigthness of the Erdős–Sós conjecture is the union of vertex-disjoint cliques of size $k-1$. This is not a connected graph and we are only aware of one result concerning $\mathrm{ex}_c(n,T)$ (but there exist results on Turán problems in connected host graphs, see e.g. [2]). We denote by P_k the path on k vertices. The value of $\mathrm{ex}_c(n,P_k)$ was determined by Kopylov, and independently by Balister, Győri, Lehel, and Schelp with the latter group also showing the uniqueness of extremal constructions.

Theorem 2 (Kopylov [8], Balister, Győri, Lehel, Schelp [1]) *If G is an n-vertex connected graph that does not contain any paths on $k+1$ vertices, then*

$$e(G) \leq \max\left\{\binom{k-1}{2} + n - k + 1, \binom{\lceil \frac{k+1}{2} \rceil}{2} + \left\lfloor \frac{k-1}{2} \right\rfloor \left(n - \left\lceil \frac{k+1}{2} \right\rceil\right)\right\}$$

holds.

We shall now present the various results obtained concerning $\mathrm{ex}_c(n,T)$. Lower bound constructions are given in Section 2 and exact determination of $\mathrm{ex}_c(n,T)$ including all trees up to 6 vertices is included in Section 3.

Our first result gathers several constructions, all based on some graph parameters, that provide lower bounds on $\mathrm{ex}_c(n,T)$. For those parameters we use the following notation.

Definition 3

- *$\ell(G)$ denotes the number of vertices in a longest path in G.*
- *$p(G)$ denotes the maximum number of vertices in a path P of G such that for all $x \in V(P)$ we have $d_G(x) \leq 2$.*
- *$\Delta(G)$ and $\delta(G)$ denote the maximum and the minimum degree in G.*
- *$\nu(G)$ denotes the number of edges in a largest matching of G.*
- *$\delta_2(T)$ denotes the smallest degree in T that is larger than 1.*
- *For a vertex $v \in V(T)$ let $m_T(v)$ be the size of largest component of $T-v$ and let $m(T) = \min\{m_T(v) : v \in V(T)\}$.*
- *For a vertex $v \in V(T)$ let $m_{T,2}(v)$ be the sum of the sizes of two largest components of $T-v$ and let $m_2(T) = \min\{m_{T,2}(v) : v \in V(T)\}$.*
- *For an edge $e = xy \in E(G)$ we write $w(e) = \min\{d_G(x), d_G(y)\}$ and define $w(G) = \max\{w(e) : e \in E(G)\}$.*

Proposition 4 *Suppose T is a tree on $k \geq 4$ vertices.*

1. $\mathrm{ex}_c(n,T) \geq \binom{\lceil \frac{\ell(T)}{2} \rceil}{2} + \lfloor \frac{\ell(T)-2}{2} \rfloor (n - \lceil \frac{\ell(T)}{2} \rceil)$.
2. $\mathrm{ex}_c(n,T) \geq (\binom{k-2p(T)-3}{2} + p(T) + 2)\lfloor \frac{n}{k-p(T)-2} \rfloor$. *Furthermore, if T contains at least two vertices of degree at least three, then* $\mathrm{ex}_c(n,T) \geq \frac{\binom{k-p(T)-1}{2}+p(T)+2}{k} n - O(k)$.

3. $\mathrm{ex}_c(n,T) \geq \lfloor \frac{n(\Delta(T)-1)}{2} \rfloor$.

4. $\mathrm{ex}_c(n,T) \geq (\nu(T)-1)(n-\nu(T)+1) + \binom{\nu(T)-1}{2}$.

5. *If T is not a star and $\delta_2(T) > 2$, then* $\mathrm{ex}_c(n,T) \geq \lfloor \frac{n-1}{k-1} \rfloor (\binom{k-2}{2} + \delta_2(T) - 1)$.

6. *If the bipartition of T consists of classes of sizes a and b with $a \leq b$, then* $\mathrm{ex}_c(n,T) \geq (a-1)(n-a+1)$.

7. *If T is not a path, then* $\mathrm{ex}_c(n,T) \geq n - 1 + \lfloor \frac{n-1}{m(T)-1} \rfloor \binom{m(T)-1}{2}$.

8. $\mathrm{ex}_c(n,T) \geq \lfloor \frac{n}{k-m_2(T)} \rfloor (1 + \binom{k-m_2(T)}{2})$.

9. $\mathrm{ex}_c(n,T) \geq (w(T)-1)(n-w(T)+1)$.

According to the Erdős–Sós conjecture, $\mathrm{ex}(n,T) = \frac{k-2}{2}n + O_k(1)$. We would like to know how much smaller $\mathrm{ex}_c(n,T)$ can be than $\mathrm{ex}(n,T)$. For any tree T we introduce

$$\gamma_T := \limsup_n \frac{2}{|T|-2} \frac{\mathrm{ex}_c(n,T)}{n}$$

where $|T|$ denotes the number of vertices in T. It is well-known that any graph with average degree at least $2d$ contains a subgraph with minimum degree at least d. Also, any tree on k vertices can be embedded to any graph with minimum degree at least k. This shows that $\gamma_T \leq 2$ for any tree T on k vertices. The Erdős–Sós conjecture would imply $\gamma_T \leq 1$.

Let $\mathcal{T}_k$ denote the set of trees on at least k vertices. We write $\gamma_k := \inf\{\gamma_T : T \in \mathcal{T}_k\}$ and $\gamma := \lim_{k\to\infty} \gamma_k$ (the limit exists as γ_k is monotone increasing).

Theorem 5 *The following upper and lower bounds hold:* $\frac{1}{3} \leq \gamma \leq \frac{2}{3}$.

Finally, we determine $\mathrm{ex}_c(n,T)$ for all trees on k vertices with $4 \leq k \leq 6$ (note that there do not exist P_3-free connected graphs), and some trees on 7 vertices. We need some notation first.

$D_{a,b}$ denotes the *double star* on $a+b+2$ vertices such that the two non-leaf vertices have degree $a+1$ and $b+1$. The *star with k leaves* is denoted by S_k. $S_{a_1,a_2,\ldots,a_j}$ with $j \geq 3$ denotes the *spider* obtained from j paths with $a_1, a_2, \ldots, a_j$ edges by identifying one endpoint of all paths. So $S_{a_1,a_2,\ldots,a_j}$ has $1 + \sum_{i=1}^{j} a_i$ vertices and maximum degree j. The only vertex of degree at least 3 is the *center* of the spider, the maximal paths starting at the center are the *legs* of the spider. M_n denotes the matching on n vertices (so if n is odd, then an isolated vertex and $\lfloor \frac{n}{2} \rfloor$ isolated edges).

For graphs H and G, their join is denoted by $H+G$, their disjoint union is denoted by $H \cup G$. For a graph H and a positive integer k, kH denotes the pairwise vertex-disjoint union of k copies of H.

The values of $\mathrm{ex}_c(n, P_{k+1})$ were determined by Theorem 2, and for $k \geq 3$, the statement $\mathrm{ex}_c(n, S_k) = \lfloor \frac{n(k-1)}{2} \rfloor$ follows from Proposition 4 (3) and that the degree-sum of an S_k-free graph is at most $n(k-1)$. So in the next theorem, we only list those trees that are neither paths nor stars. In particular, all trees have 5 or 6 vertices.

Theorem 6 *For non-star, non-path trees with 5 or 6 vertices, the following exact results are valid.*

1. *For any $T = S_{2,1,\ldots,1}$ we have* $\mathrm{ex}_c(n,T) = \lfloor \frac{n(\Delta(T)-1)}{2} \rfloor$ *if* $n \geq |T|$. *In particular,* $\mathrm{ex}_c(n, S_{2,1,1}) = n$ *if* $n \geq 5$ *and* $\mathrm{ex}_c(n, S_{2,1,1,1}) = \lfloor \frac{3n}{2} \rfloor$ *if* $n \geq 6$.

2. *We have* $\mathrm{ex}_c(n, D_{2,2}) = 2n-4$ *if* $n \geq 6$.

3. *We have* $\mathrm{ex}_c(n, S_{3,1,1}) = \lfloor \frac{3(n-1)}{2} \rfloor$ *if* $n \geq 7$ *and* $\mathrm{ex}(6, S_{3,1,1}) = 9$.

4. *We have* $\mathrm{ex}_c(n, S_{2,2,1}) = 2n-3$ *if* $n \geq 6$.

Number of vertices	Tree	$\mathrm{ex}_c(n,T)$	Construction
4	P_4	$n-1$	S_{n-1}
	S_3	n	C_n
5	P_5	n	$K_1+(K_2\cup E_{n-3})$
	S_4	$\lfloor\frac{3n}{2}\rfloor$	(nearly) 3-regular
	$S_{2,1,1}$	n	C_n
6	P_6	$2n-3$	K_2+E_{n-2}
	S_5	$2n$	4-regular
	$S_{2,1,1,1}$	$\lfloor\frac{3n}{2}\rfloor$	(nearly) 3-regular
	$S_{2,2,1}$	$2n-3$	K_2+E_{n-2}
	$S_{3,1,1}$	$\lfloor\frac{3(n-1)}{2}\rfloor$	K_1+M_{n-1}
	$D_{2,2}$	$2n-4$	$K_{2,n-2}$

Table 1: The value of $\mathrm{ex}_c(n,T)$ for all trees up to 6 vertices

Tree	$\mathrm{ex}_c(n,T)$	Construction	Tree	$\mathrm{ex}_c(n,T)$	Construction
S_6	$\lfloor\frac{5n}{2}\rfloor$	(nearly) 5-regular	P_7	$2n-2$	$K_2+(E_{n-4}\cup K_2)$
$S_{4,1,1}$	$\geq 2n-3$	K_2+E_{n-2}	$S_{3,2,1}$	$2n-3$	K_2+E_{n-2}
$S_{3,1,1,1}$	$\lfloor\frac{3n}{2}\rfloor$	(nearly) 3-regular	$S_{2,1,1,1,1}$	$2n$	4-regular
$S_{2,2,2}$	$2n-2$	$K_2+(E_{n-4}\cup K_2)$	$S_{2,2,1,1}$	$\geq 2n-3$	K_2+E_{n-2}
$D^*_{2,2}$	$2n-3$	K_2+E_{n-2}	$D_{2,3}$	$\geq 2n-4$	$K_{2,n-2}$
$SD_{2,2}$	$\geq \frac{13n}{7}-O(1)$	Prop. 4 (2)	$D_{2,3}$	$\geq 2n-2$ if $6\vert n-1$	Prop 4 (5)

Table 2: Exact values and lower bounds on $\mathrm{ex}_c(n,T)$ for trees with 7 vertices

Let $D^*_{2,2}$ be the tree obtained from $D_{2,2}$ by attaching a leaf to one leaf of $D_{2,2}$.

Theorem 7 *We have $ex_c(D^*_{2,2})=2n-3$ for all $n\geq 7$, and $ex_c(D^*_{2,2})=\binom{n}{2}$ for $1\leq n\leq 6$.*

Theorem 8 *We have $ex_c(S_{2,2,2})=2n-2$ for all $n\geq 7$, and $ex_c(S_{2,2,2})=\binom{n}{2}$ for $1\leq n\leq 6$.*

Theorem 9 *We have $ex_c(S_{3,2,1})=2n-3$ for all $n\geq 7$, and $ex_c(S_{3,2,1})=\binom{n}{2}$ for $1\leq n\leq 6$.*

Theorem 10 *For any $T=S_{3,1,\ldots,1}$ with $\Delta(T)\geq 4$, we have $\mathrm{ex}_c(n,T)=\lfloor\frac{(\Delta(T)-1)n}{2}\rfloor$ if n is large enough.*

For a better overview, we include tables with previous results, our results and open cases for trees up to 7 vertices. $SD_{2,2}$ denotes the tree on 7 vertices obtained from the double star $D_{2,2}$ by subdividing the edge connecting its two centers.

References

[1] P.N. Balister, E, Győri, J. Lehel, R.H. Schelp, Connected graphs without long paths, *Discrete Mathematics*, **308(19)** (2008), 4487–4494.

[2] N. Bougard, G. Joret, Turán's theorem and k-connected graphs. *Journal of Graph Theory*, **58(1)** (2008), 1–13.

[3] N. Bushaw, N. Kettle, Turán numbers of multiple paths and equibipartite forests. *Combinatorics, Probability & Computing*, **20(6)** (2011), 837–853.

[4] P. Erdős, Extremal problems in graph theory. In *Theory of graphs and its applications, Proc. Sympos. Smolenice* (1964), 29–36.

[5] P. Erdős, M. Simonovits, A limit theorem in graph theory. *Studia Sci. Math. Hungar.*, **1** (1966), 51–57.

[6] P. Erdős, A.H. Stone, On the structure of linear graphs. *Bull. Amer. Math. Soc. 52* (1946), 1087–1091.

[7] Z. Füredi, M. Simonovits, The history of degenerate (bipartite) extremal graph problems, In *Erdős Centennial*, Bolyai Soc. Math. Stud., 25, János Bolyai Math. Soc., Budapest (2013), 169–264.

[8] G.N. Kopylov, On maximal paths and cycles in a graph. *Doklady Akademii Nauk SSSR*, **234(1)** (1977), 19–21.

[9] Y. Lan, T. Li, Y. Shi, J. Tu, The Turán number of star forests. *Applied Mathematics and Computation*, **348** (2019), 270–274.

[10] B. Lidický, H. Liu, C. Palmer, On the Turán number of forests. *The Electronic Journal of Combinatorics*, **20(2)** (2013), #P62.

[11] M. Stein, Tree containment and degree conditions. In: Raigorodskii, A.M., Rassias, M.T. (eds) *Discrete Mathematics and Applications*. Springer Optimization and Its Applications, vol. 165 (2020), 459–486. Springer, Cham.

[12] L.T. Yuan, X.D. Zhang, The Turán number of disjoint copies of paths. *Discrete Mathematics*, **340(2)** (2017), 132–139.

[13] L.P. Zhang, L. Wang, The Turán numbers of special forests. *Graphs and Combinatorics*, **38(3)** (2022), #84.

On a matrix representation of a sequence of chordal graphs

Dániel Pfeifer

Department of Differential Equations
Budapest University of
Technology and Economics, Hungary
pfeiferd@math.bme.hu

Edith Alice Kovács

Department of Differential Equations
Budapest University of
Technology and Economics, Hungary
kovacsea@math.bme.hu

Abstract: Chordal graphs have useful applications in various fields. A special case of them, the cherry tree graphs are applied in probability theory in order to define the so-called cherry tree distributions. For discrete probability distributions, the cherry tree graph is sufficient for their definition. In the continuous case, the so-called vine structure is needed, which can be defined as a sequence of so-called regular cherry trees. In this paper, we give a matrix representation of such a sequence of cherry trees and highlight the advantages of this representation.

Keywords: Chordal graphs, Junction trees, Cherry trees, Matrix representation

1 Introduction

The cherry tree graph structure is a basic concept in estimating best fitting probability distributions, based on given order marginals [1] [2]. In this paper we discuss problems with regard to their graph structures and their encoding into a matrix.

In the continuous setting, to obtain a formula for the multivariate probability density approximations, one can use the concept of copulas, and an even more flexible type of multidimensional copulas, the vine copulas [3]. The formula of a vine copula corresponds to a so-called vine graph structure. In this paper we give an overview on how a vine graph structure is connected to chordal graphs, and also give an algorithm, for encoding these structures (sequence of chordal graphs) into a matrix.

The paper is structured as follows: In the preliminary part we introduce cherry trees starting from the well known definitions in graph theory. This part also contains a brief introduction of the original vine structure representation. The third part highlights how special cherry trees are used to build trees of the vine structure, how the vine structure is related to chordal graphs, and why cherry trees are more general than the cluster trees used in the vine structures.

In the fourth part an algorithm is given for the encoding of a vine structure into a matrix. Here we emphasize the advantages of the method presented.

2 Preliminaries

In this part firstly we introduce junction trees, and how they relate to chordal graphs. Then as a special case of them, we define cherry trees which are a central concept of our research. In the second subsection, we give the definition of vine structure, which we express with a sequence of cherry trees.

2.1 Graph structures

Consider a graph $G = G(V, E)$ with the set of vertices $V = \{v_1, \ldots, v_n\}$ and the set of undirected edges E.

Definition 1 *A graph $G(V, E)$ is said to be chordal when every cycle of length 4 or more has a chord (an edge joining two non-consecutive vertices of the cycle). (See the leftmost graph on Figure 1.)*

Definition 2 *Given a graph $G = (V, E)$ and a node $v \in V$, the neighbourhood of v is defined as*

$$Ne(v) = \{w \in V | (v, w) \in E\}$$

or all the nodes in G that connect to v.

Definition 3 *The perfect elimination ordering of a graph $G(V, E)$ is an ordering $r_1, \ldots, r_n$ of its vertices $v_1, \ldots, v_n$ such that for all $i \in \{1, \ldots, n\}$: $Ne(r_i) \cap \{r_{i+1}, \ldots, r_n\}$ is a clique in the remaining subgraph of $G(r_{i+1}, \ldots, r_n)$.*

Not all graphs G have a perfect elimination ordering. The following theorem gives a necessary and sufficient condition for this property.

Theorem 4 *G is chordal if and only if G has a perfect elimination ordering. [4]*

Basic concepts and properties of chordal graphs can be found in [5] and [6].
We need the following basic definition.

Definition 5 *A maximal clique of a graph G is a clique which is not a subgraph of any other clique of G. We will call these clusters.*

We introduce now the general concept of intersection graph. [7]
Consider a family of non-empty sets.

Definition 6 *The intersection graph of this family is obtained by representing each set by a vertex, two vertices are connected by an edge if and only if the corresponding sets intersect. (See the central graph on Figure 1.)*

The problem of characterizing the intersection graph of a family of sets having a defined topological pattern is of great interest in different domains (for example interval graphs).

Let us suppose we have a graph G. We denote the set of clusters (maximal cliques) by $\mu(G)$ and the set of clusters which contain the vertex $v \in V$ by $\mu_v(G)$.

Theorem 7 *A graph $G(V)$ is a subtree graph if and only if there exists a tree T whose set of vertices is $\mu(G)$, so that, for every $v \in V$,$T(\mu_v(G))$ is connected. [7]*

Theorem 8 *G is a subtree graph if and only if it is chordal graph. [7]*

The next result was inspired by a definition given in [8].

Definition 9 *The weighted cluster-intersection graph of a chordal graph is an intersection graph with the set of vertices defined by $\mu(G)$, whose edges connect the non-disjoint clusters. The weight of the edges is given by the cardinality of the elements of the intersection of the clusters it connects.*

Theorem 10 *A maximum weighted spanning tree of the weighted cluster-intersection graph determines the chordal graph G.*

Definition 11 *Every maximum-weight spanning tree of the cluster-intersection graph of G is called a cluster tree of G. (See the rightmost graph on Figure 1.)*

We can define a junction tree in the following way:

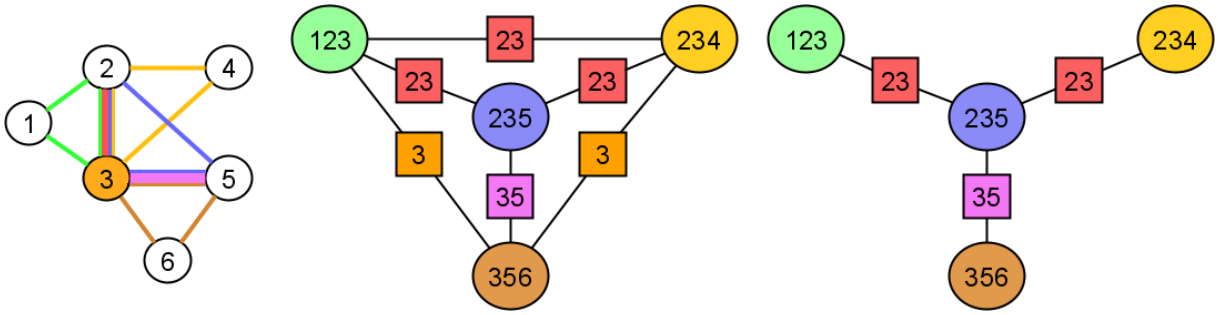


Figure 1: Example for the construction of chordal to junction (cherry) tree. On the left, the maximal clusters (here, of size 3) are denoted by different colors. In the middle, we changed to the cluster notation. The intersections (separators) are shown in squares. If each edge is weighted by the cardinality of its separator, then a maximal weight spanning tree becomes a junction (cherry) tree. One potential outcome is on the right.

Definition 12 *A junction tree is obtained when the clusters are the maximal cliques of a chordal graph, and the edges, called separators, are given by the set of elements in the intersection of the endpoint clusters of the given edge.*

For any junction tree the following property, called running intersection property holds.

Theorem 13 *If an element is contained in two different clusters of a junction tree, then it is contained in all separators and clusters on the path between the two clusters.*

This theorem follows straightforward from Theorem 8.

Now we can define the k-th order cherry tree, introduced in (Szántai-Bukszar, Kovacs-Szántai) [2] in a constructive way, as a special junction tree.

Definition 14 *A k-order cherry tree is a special junction tree, in which all clusters consist of k elements, and all separators consist of $k-1$ elements. (See the rightmost graph on Figure 1.)*

2.2 Connection between chordal graphs and probabilistic graphical models

In this section we give a brief insight into how junction trees are related to probabilistic graphical models.

Let us consider a random vector $\mathbf{X} = (X_1, \ldots, X_n)^T$ with probability distribution $P(\mathbf{X})$ and let us denote the set of indices by $V = \{1, \ldots, n\}$.

Let us now define a junction tree over the set of indices and let us denote the set of clusters by $\mathcal{C}$ and the set of separators $\mathcal{S}$.

The following formula gives a valid probability distribution:

$$P_{\text{J-Tree}}(\mathbf{X}) = \frac{\prod\limits_{C \in \mathcal{C}} P(\mathbf{X}_C)}{\prod\limits_{S \in \mathcal{S}} [P(\mathbf{X}_S)]^{(v_S - 1)}} \tag{1}$$

where we use the notation $P(\mathbf{X}_A)$ for a marginal probability distribution of $\mathbf{X}$, having indices in $A \subset V$, and v_S denotes how many times S is a separator between two clusters.

The following question arises naturally. How can we find a good approximation of form (1) for $P(\mathbf{X})$.

It is proved in [9] that the best approximation of $P(\mathbf{X})$ by a k-width junction tree probability distribution (largest cluster is of size k) over V is given by the best fitting k-width cherry tree probability distribution.

In the discrete case, the problem can be approached in a greedy way (without a guarantee of optimality). In the continuous case, we need to use the concept of vine copulas. In [1] [2] it is proved, that the vine structure can be expressed as a sequence of special cherry trees. The vine copula approach is appealing from multiple viewpoints: copula makes it possible to model the dependence structure and the marginal probability distributions separately. This is why we are concerned with encoding these cherry trees into a matrix. The next sections will deal with problems related to this scope.

2.3 Vine structures

In 2001, Bedford T. and R. M. Cooke showed that a multidimensional copula can be split into a special product, whose elements are pair-copulas and conditional pair-copula p.d.f.'s [10]. This formula was assigned to a specific graph structure, made up of a sequence of trees. Bedford T. and R.M. Cooke called this special structure a "vine", which was explained in detail in their 2002 paper [11]. We will refer to this vine structure as the *classical vine structure*. The vine structure is a special sequence of cluster trees. For an informative look at vines with applications, see [3].

The first cluster tree is an ordinary tree, whose vertices are the indices of the variables $(1, \ldots, n)$. The graph structure on n variables contains a total of $n-1$ so-called "cluster trees".

We denote the k'th cluster tree in the sequence by T_k. These cluster trees are defined by using the following rules:

- T_1 is any spanning tree on vertices $1, \ldots, n$.
- The T_k cluster tree is defined by $n-k+1$ sets (clusters). Each cluster in T_k contains exactly k elements: $\{a_1, \ldots, a_k\}$. To keep it simple, when drawing the tree, we will omit the set notation.
- If $A = a_1 \ldots a_k$ and $B = b_1 \ldots b_k$ are two connected clusters, then the label of the edge running between them should be $D|S$, where $D = (A \cup B) \setminus (A \cap B)$, also known as the symmetric difference of sets A and B, and $S = A \cap B$, or the intersection of sets A and B. (For example if $A = 235$ and $B = 236$, then $D|S = 56|23$. If $S = \emptyset$, then we omit S, and also omit the line in front of it.)
- The clusters of $T_{k+1}, k \geq 1$ contain exactly the same elements as the edge labels of T_k. (For example if the edge labels of T_3 are 14|23 and 25|34, then the clusters of T_4 are 1423 and 2534.)
- Two clusters (A and B) can only be connected if $|(A \cup B) \setminus (A \cap B)| = 2$. Because of this, in the label of the connecting edge $D|S$, D will contain exactly 2 elements.

Any T_k tree defined by the previous rules is a cluster tree, however for simplicity, and if it does not cause confusion, we will sometimes refer to them as trees.

Vine structures can be drawn easily (see Figure 2). We start from a spanning tree, and then we build every consecutive tree from the previous one, by copying the edge indices of T_k into the clusters of T_{k+1}, and connecting up the resulting clusters so that they form a cluster tree. We also have to make sure to only connect clusters where the size of the symmetric difference of the labels is exactly 2.

Therefore all clusters of the k'th tree will contain $k+1$ elements, and each edge will contain 2 elements before the condition line and $k-1$ elements after.

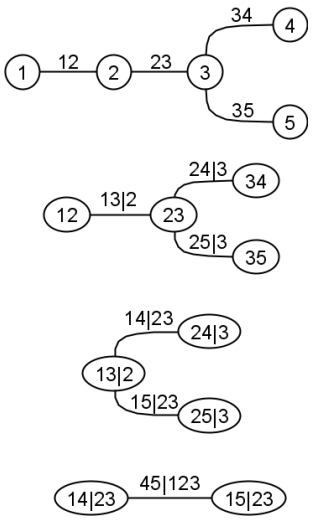


Figure 2: Example of a classical represented vine structure on 5 variables

3 Vine representations

In this part, we show how a vine structure can be expressed in a more compact way by using a sequence of special cherry trees, then we discuss their connection to chordal graphs.

3.1 Vine structure representation using a cherry tree sequence

We will now show how the cherry tree sequence, introduced by Edith Kovács and Tamás Szántai [12] [1] can be used in the description of a vine structure.

Now we will show how a cluster tree of a vine structure corresponds to a cherry tree:

- The clusters of the cherry tree correspond to the clusters of the vine cluster tree. We will not separate the part before and after the condition here.
- The indices behind the separation on the edges of the classical vine, correspond to the intersection of two connected clusters, and are assigned to the separators of the cherry tree, and are denoted in a rectangle.
- In order to define uniquely the cherry tree assigned to the cluster tree, each separator has to be represented in the cherry tree, even if it appears multiple times.

It is important to highlight that cherry trees are more general structures than cluster trees defining the vines. There exist cherry trees which cannot be assigned to a tree in a vine.

Therefore we introduce the following important definition:

Definition 15 *A k'th order cherry tree with the property that all its separators form a cherry tree of order $k-1$ is called regular cherry tree.*

As for an example, refer to the tree on Figure 3. Here, the sets 123 and 236 are the separators, but there are three clusters that join to 123, since in the original graph these three clusters each have a connecting edge with a conditioning set of 123.

The construction from a vine to a regular cherry tree sequence is unique forwards and backwards.

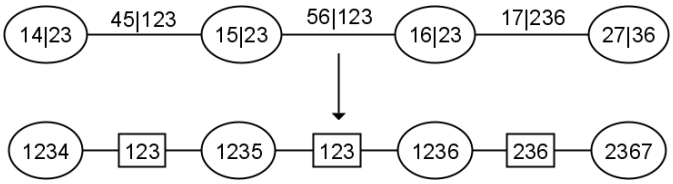


Figure 3: Creating a regular cherry tree from a cluster tree of a vine structure

Theorem 16 *(Running intersection property)*
For all cluster trees in the vine structure, if A and B are different clusters in the same tree, containing an element s (before or after the condition), then s appears in all clusters and all edges on the path between A and B.

Proof: The proof is straightforward, because of the correspondence to cherry tree which is a special type of junction tree, and therefore the running intersection property holds. □

On Figure 4, one can see a cherry tree which is not regular. (No matter how we try to form a cherry tree out of the clusters 123, 124 and 134, the running intersection property will fail.)

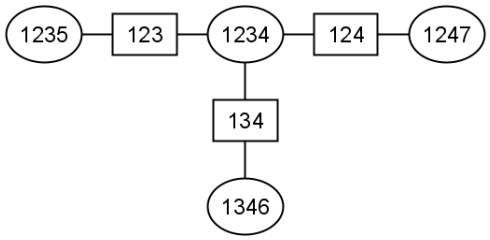


Figure 4: A 3rd order cherry tree which is not a regular cherry tree.

Cherry trees in vines are regular cherry trees. This is a consequence of the way they are built.

3.2 Vine structure representation using a chordal graph sequence

Chordal graph representations can also be used to represent a vine. This section will serve as the basis of Algorithm 1, where the perfect elimination ordering property of chordal graphs will be used heavily. We will show how a chordal graph sequence can be constructed from a cherry tree sequence (which is a characterization of a vine structure).

Based on Definition 12 we know that cherry trees are special junction trees, therefore the clusters are maximal cliques.

So the separator (intersection) of two clusters of size k is a size $k-1$ clique. (See Figure 5 for an example with $k = 4$.)

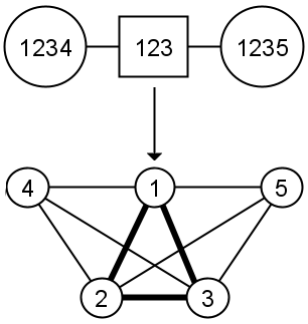


Figure 5: The chordal graph representation of two connected clusters. The separator, 123 forms a complete graph (denoted by thicker lines), while the clusters 1234 and 1235 also form complete graphs.

Theorem 17 *Given a vine on n variables with cluster trees $T_1, \ldots, T_{n-1}$ in chordal graph representation; there exists an ordering of its variables $1, \ldots, n$, such that it is a perfect elimination ordering for every tree in $T_1, \ldots, T_{n-1}$.*

Proof: We will give a constructive proof to find this perfect elimination ordering.

Observation: Once an index became apart of a separator, it will remain in a separator in all subsequent cluster trees. This follows from the definition of vine structures.

Starting from T_{n-1}, the first two elements of the perfect elimination ordering should be the symmetric difference of the two clusters. Their neighbourhood is the separator of the two clusters, which forms a clique. According to the Observation, these elements could not have been in a separator in any previous cluster tree, so their neighborhood was always a separator of two clusters, which again forms a clique.

Moving on to T_{n-2}. If there are more indices that are not part of a separator, add them to the perfect elimination ordering. Once again, according to the Observation, they could not have come from a separator, so their neighborhood in any previous tree is an intersection of two clusters, which forms a clique.

And so on, moving backwards, add all indices in any order that are not part of a separator. According to the previous argument, this will always fulfil the conditions of a perfect elimination ordering.

After finishing, add all the remaining indices in any order. These have always formed a clique in every cluster tree, so any ordering of their indices is a valid perfect elimination ordering, thus the graph is chordal. □

We will conclude this section with an important Definition that will be necessary for Section 4:

Definition 18 *We call perfect elimination ordering of a vine structure any perfect elimination ordering of all of its cluster trees. According to Theorem 17., such an ordering always exists.*

In this section we have shown that the vine structure can be represented as a sequence of cherry trees and a sequence of chordal graphs. We have also proved an important theorem regarding to the perfect elimination ordering, which will be used in the next section.

4 Matrix representation of a vine structure

In this section we give an algorithm which encodes a vine structure given by a sequence of regular cherry trees into a matrix. The vine matrix building algorithm described in the O. M. Nápoles et al paper [13] encodes a vine structure into a lower diagonal matrix column by column. O. M. Nápoles' algorithm can be used only in cases where the vine structure contains all trees starting from the regular tree to

the complete graph. This algorithm is not able to encode the structure if the input is a regular cherry tree of a given level and cannot be used if the vine structure is constructed tree by tree based on some optimization criteria.

In the first subsection we give the algorithm of the matrix encoding of a vine structure defined by regular cherry trees. In the second subsection we show, how a given regular cherry tree can be achieved as a sequence of trees which can be encoded in a matrix.

4.1 Cherry tree based vine structure encoding matrix

The algorithm which we present here encodes a vine structure given by a sequence of regular cherry trees into lower triangulated matrix. This algorithm builds the matrix row by row. This enables us to encode also truncated vine structures.

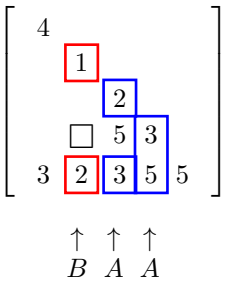
Algorithm 1 Row-wise vine matrix building method using a cherry tree sequence

Input: The trees in the vine structure with an adjacency list

Let $r_1, \ldots, r_n$ be one of the vine's perfect elimination orderings defined in Definition 18. ▷ Remark 19
for $j = n$ to 1 **do**
  $m_{j,j} := r_j$
  $m_{n,j} :=$ The node in T_1, where r_j connects to one of $\{r_{j+1}, \ldots, r_n\}$ ▷ Theorem 20
end for
for $i = n - 1$ to 2 **do**
  for $j = i - 1$ to 1 **do**
    for $k = j + 1$ to i **do**
      $A := \{m_{k,k}\} \cup \{m_{i+1,k}, \ldots, m_{n,k}\}$
      $B := \{m_{j,j}\} \cup \{m_{i+1,j}, \ldots, m_{n,j}\}$
      if the clusters A and B are connected in T_{n-i+1} **then** ▷ Theorem 21
        $m_{i,j} =$ The single element of $A \setminus B$ ▷ Theorem 21
      end if
    end for
  end for
end for
All other elements of M are 0.

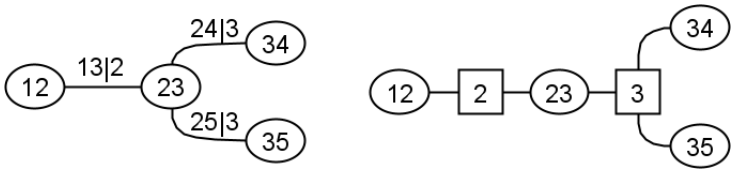
Output: The M matrix

Remark 19 *Firstly let us examine why the perfect elimination order is needed here. We will once again work with the vine structure shown on Figure 2, and assume that this is where we currently are in the algorithm:*



Where the upcoming element to fill is □*. For it to be unique,* $A \setminus B$ *has to contain exactly one element. Here* $B = \{1, 2\}$ *and for* A*, we have the following two options:* $\{2, 3\}$ *and* $\{3, 5\}$*. Clearly,* $A \setminus B$ *has exactly one element if* $A = \{2, 3\}$*, and then* $A \setminus B = \{3\}$*, so the new element is* $\square = 3$*.*

The only way we can guarantee that the difference of these sets contains exactly one element is if the clusters A and B are connected in their cluster tree. We can only connect them if the symmetric difference of A and B contains 2 elements, meaning they differ in one/one elements, $A \setminus B$ and $B \setminus A$. Looking at the vine structure it is easy to read that 12 is not connected to 35 in T_2, but it is connected to 23:



With the perfect elimination ordering of the vine; we get a reordeing of the indices $1, \ldots, n$ so that this holds true. For every tree T_k, and every index in the perfect elimination ordering r_j, $Ne(r_j) \cap \{r_{j+1}, \ldots, r_{n-k+1}\}$ forms a complete graph, so r_j is definitely connected to at least one cluster in T_k containing at least one of $\{r_{j+1}, \ldots, r_{n-k+1}\}$. This way, there is at least one cluster A for every cluster B such that $A \setminus B$ contains exactly one element.

For now it may be unclear why such clusters even appear in the trees of the vine structure. This will be proven in Theorem 21.

Theorem 20 *In Algorithm 1, in the construction of the main diagonal and the bottom row (in the first for-loop), r_j connects to exactly one element out of $\{r_{j+1}, \ldots, r_n\}$ in T_1.*

Proof: It connects to at least one node, since applying the perfect elimination ordering the same way as in 19 for $k = 1$, we get that r_j connects to at least one node out of $\{r_{j+1}, \ldots, r_{n-k+1}\} = \{r_{j+1}, \ldots, r_n\}$.

But it cannot connect to any more, since if it did (say it connects to both r_x and r_y), then there would exist an $r_j r_x ... r_y r_j$ cycle. (Since the remaining portion of the graph is connected, there exists a path between r_x to r_y.)

And if $|\{r_{j+1}, \ldots, r_{n-k+1}\}| = 1$, for example $\{r_{j+1}, \ldots, r_{n-k+1}\} = \{r_x\}$, then r_x has to once again connect to exactly one element in the remaining portion of the tree, since the tree is connected. □

Theorem 21 *The clusters $A := \{m_{k,k}\} \cup \{m_{i+1,k}, \ldots, m_{n,k}\}$ and $B := \{m_{j,j}\} \cup \{m_{i+1,j}, \ldots, m_{n,j}\}$ appear in T_{n-i+1}, and if they are connected, then $A \setminus B$ contains exactly one element.*

Proof: Let us start with $i = n - 1$, just as Algorithm 1 does. Then, since all $m_{j,j} = r_j$ elements appear in T_1, and we chose $m_{n,j}$ to be the element that connects to $m_{j,j}$ in T_1, the edge label between the two nodes in T_1 is $m_{j,j} m_{n,j}$. These are exactly the elements that appear in the clusters of T_2, so the first half of the theorem is true for $i = n - 1$.

Using induction, let us now decrease the value of i in every step, and prove the theorem for T_{n-i+1} using what we have proven for T_{n-i}. (For $i = n - 1$, we have proven the theorem for T_2 using T_1, now we will prove it in order from $T_3, T_4, \ldots$, etc. using the previous tree in each case.)

T_{n-i} contains clusters of size i, so using the induction hypothesis we have shown that the clusters of form $A' = \{m_{k,k}\} \cup \{m_{i+2,k}, \ldots, m_{n,k}\}$, where $k \in \{j+1, \ldots, i\}$ all appear in T_{n-i}. Let A' be one of the previous sets for a fixed k, and $B' := \{m_{j,j}\} \cup \{m_{i+2,j}, \ldots, m_{n,j}\}$, which also appears in T_{n-i} using the induction hypothesis. In order for the algorithm to deal with these sets, they have to be connected. But then, using the properties of the vine structure, they have to differ in exactly one/one element. So let us now rewrite them as $A' = \{s, p_1, \ldots, p_{n-i-1}\}$ and $B' = \{t, p_1, \ldots, p_{n-i-1}\}$. Then the label of the edge running between them is $st|p_1 \ldots p_{n-i-1}$, so in the following tree, T_{n-i+1}, the cluster $\{s, t, p_1, \ldots, p_{n-i-1}\}$ will definitely appear. Let us call this cluster $C \in T_{n-i+1}$.

Now let us examine, what we would insert into the matrix M into position $m_{i+1,j}$. According to Algorithm 1, $A' \setminus B'$ would be inserted there. This element, according to the above expansion, is s.

With this, we now actually know what the original B set was. Since

$$B = \{m_{j,j}\} \cup \{m_{i+1,j}, \ldots, m_{n,j}\} = \{m_{j,j}\} \cup \{s, m_{i+2,j}, \ldots, m_{n,j}\} =$$
$$= \{s\} \cup (\{m_{j,j}\} \cup \{m_{i+2,j} \ldots, m_{n,j}\}) = \{s\} \cup B' = \{s\} \cup \{t, p_1, \ldots, p_{n-i-1}\} = C$$

So the set B, as a cluster, indeed appears in the original tree, it will be the cluster of T_{n-i+1} that was obtained from the edge between A' and B' in T_{n-i}.

Since set A is a special B-type set, which was obtained by taken a different column into account (the j'th instead of the k'th), the theorem is also true for A, no matter what value k takes.

Because of induction, the theorem will be true for all j indices, so the sets A and B, which were obtained from the matrix, indeed appear in the original vine structure as clusters.

The second statement of the theorem is that the set $A \setminus B$ contains exactly one element. However, this is almost trivial, since we can only get to this branch of the Algorithm is A and B are connected in T_{n-i+1}. As always, the label of the edge running between them is $s_1 s_2 | t_1 \ldots t_{n-i-1}$, where A and B differ in exactly one/one element, s_1 and s_2. So $A \setminus B$ is either $\{s_1\}$ or $\{s_2\}$, definitely a one-element set. □

4.2 Achieving a k'th order cherry tree with a cherry tree sequence

The problem to be solved can be split into two subproblems, the regular and irregular case.

Given a k'th order cherry tree (T_k), if it is regular, then we can find a sequence of regular cherry trees starting from a T_1 tree with clusters of size 1, using vine steps (outlined in Section 3.1, moving from T_j to T_{j+1}) so that the end result is the given k'th order cherry tree.

In the regular case, we refer to the constructive algorithm introduced by E. Kovács and T. Szántai in [1]. The input of the algorithm is a regular cherry tree, and its output is a vine structure, with the highest level being the regular cherry tree structure. This result is easily encoded in a matrix.

For cases in which the k'th cherry tree does not satisfy the regularity condition, we make a vine step (outlined in Section 3.1) and obtain a $k+1$'st order cherry tree which is regular [1]. Then a vine structure can be found corresponding to this, like earlier, which can be encoded in a matrix.

From a graph point of view, this single vine step adds a couple more edges to the input cherry tree (exactly as many as there are separators in tree T_k) and obtain a $k+1$'st order regular cherry tree.

5 Conclusion

The paper presents how vine structures are related to chordal graphs an cherry trees, without detailing the probabilistic background which inspired this approach. By using cherry trees we gave a method for encoding a vine structure into a matrix. The matrix filled in this way is able to describe a vine structure with a given regular cherry tree as the highest tree. In our opinion these representations can be used for a variety of optimization problems.

References

[1] Edith Kovács and Tamás Szántai On the connection between cherry-tree copulas and truncated R-vine copulas, *Kybernetika* **53 (3)**, Pages **437–460** (2017)

[2] Edith Kovács and Tamás Szántai, Hypergraphs in the characterization of regular vine copula structures, *Proc. 13th International Conference on Mathematics and its Applications, Timisoara (arXiv:1604.02652)* , Pages **335–344** (2012)

[3] Aas, Kjersti and Czado, Claudia and Frigessi, Arnoldo and Bakken, Henrik, Pair-copula constructions of multiple dependence, *Insurance: Mathematics and economics* **44 (2)**, Pages **182–198** (2009)

[4] Rose, Donald J, Triangulated graphs and the elimination process, *Journal of Mathematical Analysis and Applications* **32 (3)**, Pages **597–609** (1970)

[5] Golumbic, Martin Charles, Algorithmic graph theory and perfect graphs (2004)

[6] Blair, Jean RS and Peyton, Barry, An introduction to chordal graphs and clique trees, *Graph theory and sparse matrix computation*, Pages **1–29** (1993)

[7] Gavril, Fănică, The intersection graphs of subtrees in trees are exactly the chordal graphs, *Journal of Combinatorial Theory, Series B* **16 (1)**, Pages **47–56** (1974)

[8] Thomas, Alun and Green, Peter J, Enumerating the junction trees of a decomposable graph, *Journal of Computational and Graphical Statistics* **18 (4)**, Pages **930–940** (2009)

[9] Tamás Szántai and Edith Kovács, Hypergraphs as a mean of discovering the dependence structure of a discrete multivariate probability distribution., *Annals of Operations Research* **193 (2012)**, Pages **71-90**

[10] Bedford, T. and Cooke, R.M., Probability density decomposition for conditionally dependent random variables modeled by vines, *Annals of Mathematics and Atrificial Intelligence* **32**, Pages **245–268** (2001)

[11] Bedford, T. and Cooke, R.M., Vines—a new graphical model for dependent random variables, *Annals of Statistics* **30 (4)**, Pages **1031–1068** (2002)

[12] Edith Kovács and Tamás Szántai, On the approximation of a discrete multivariate probability distribution using the new concept of t-cherry junction tree, *Coping with Uncertainty*, Pages **39–56** (2010)

[13] Nápoles, O Morales, Bayesian belief nets and vines in aviation safety and other applications, *Delft: TU* (2009)

Color-avoiding connected spanning subgraphs with minimum number of edges

József Pintér[1]

Department of Stochastics
Budapest University of Technology and Economics,
ELKH-BME Stochastics Research Group
Budapest, Hungary
pinterj@math.bme.hu

Kitti Varga[2]

ELKH-ELTE Egerváry Research Group,
MTA-ELTE Matroid Optimization Research Group
Budapest, Hungary
vkitti@math.bme.hu

Abstract: We call a (not necessarily properly) edge-colored graph edge-color-avoiding connected if after the removal of edges of any single color, the graph remains connected. In this article, we investigate the problem of determining the maximum number of edges that can be removed such that the graph remains edge-color-avoiding connected. First, we prove that this problem is NP-hard, then we give a polynomial-time approximation algorithm for it. To analyze the approximation factor of the algorithm, we determine the minimum number of edges of edge-color-avoiding connected graphs on a given number of vertices and with a given number of colors. Furthermore, we also consider a generalization of this problem to matroids.

Keywords: approximation algorithms, color-avoiding connectivity, complexity, matroids, spanning subgraphs

1 Introduction

The robustness of networks against random errors and targeted attacks has attracted a great deal of research interest. The robustness of a network refers to its capacity to maintain some degree of connectivity after the removal of some edges or vertices of the network.

Although the standard frameworks of error or attack tolerance in complex networks can be really useful in industrial practices, we can develop a more efficient framework if we take into account that some parts of the network might share some vulnerabilities. A characteristic example is the case of public transport networks, where the edges of the underlying graph are colored according to the mode of transportation such as rail, road, ship or air transport. Experience shows that excessive snowing has a greater impact on the railway than on underground transportation. In extreme cases, these weather conditions might even paralyze the whole railway traffic, which we can think of (from a network theoretical point of view) that all the edges corresponding to railway transportation disappear from the network. Thus it is useful to know which vertices in the network are available from each other without using any edges corresponding to the railway transportation. In this manner, we can consider the network reliable if even after the elimination of any single mode of transportation, the whole or a significant part of the network remains connected. Another example is the case of communication networks where the vertices represent routers, which are colored according to which country the corresponding router is registered to. If, for safety reasons, we want to ensure that no country can intercept our message, then we need multiple paths in

[1]Research is supported by Ministry of Culture and Innovation and the National Research, Development and Innovation Office within the framework of the Artificial Intelligence National Laboratory Programme.

[2]Research is supported by the Hungarian National Research, Development and Innovation Office – NKFIH, grant number FK128673.

the network between the sender and the receiver such that each country is avoided in at least one of these paths, and send our message divided into many parts through these paths.

These concepts were introduced as color-avoiding connectivity, first for vertex-colored graphs by Krause et al. [9] in 2016 with the motivation to develop a framework which can treat the heterogeneity of multiple vulnerable classes of vertices, and they demonstrated how this can be exploited to maintain functionality of a complex network by utilizing multiple paths, mostly on communication networks. Krause et al. extended this original theory in [10]. They analyzed how the color frequencies affect the robustness of the networks. For unequal color frequencies, they found that the colors with the largest frequencies control vastly the robustness of the network, and colors of small frequency only play a little role. In [7], color-avoiding connectivity was further extended from vertex-colored graphs to edge-colored ones.

Giusfredi and Bagnoli investigated color-avoiding percolation in diluted lattices [5] and also showed that color-avoiding connectivity can be formulated as a self-organized critical problem, in which the asymptotic phase space can be obtained in one simulation [4]. Ráth et al. [17] investigated the color-avoiding bond percolation of edge-colored Erdős–Rényi random graphs. They analyzed the fraction of vertices contained in the giant edge-color-avoiding connected component and proved that its limit can be expressed in terms of probabilities associated to edge-colored branching process trees. The work [13] of Lichev and Schapira includes some simplification and generalization of these results as well as some finer results on the size of the largest edge-color-avoiding connected component. Lichev also described the phase transition of the largest edge-color-avoiding connected component between the supercritical and the intermediate regime [12].

Molontay and Varga [14] investigated the computational complexity of finding the color-avoiding connected components of a graph. They also generalized the concept of color-avoiding connectivity by making the vertices or edges more vulnerable by assigning a list of colors to them.

A similar concept called courteous edge-coloring was studied by DeVos et al. [1] in 2006. Graphs with 1-courteous edge-colorings are exactly the edge-color-avoiding connected graphs. In that article, they gave interesting upper bounds on the number of colors needed to courteously color an arbitrary graph.

When operating a network, we might want to reduce the maintenance cost while retaining some desired properties of the network. In this work, we investigate the problem of finding edge-color-avoiding connected spanning subgraphs with minimum number of edges in an edge-color-avoiding connected graph. We also generalize this problem to matroids. First, we prove that these problems are NP-hard, then we present polynomial-time approximation algorithms for them.

2 Edge-color-avoiding connected graphs and courteously colored matroids

In this article, we study edge-color-avoiding connected graphs and courteously colored matroids. First, we recall some important definitions and notation.

The set of positive integers is denoted by $\mathbb{Z}_+$. For two sets X and Y, the *set difference* of X and Y is denoted by $X - Y$. Given a graph $G = (V, E)$ and a subset of edges $E' \subseteq E$, let $G - E'$ denote the graph that is obtained from G by deleting the edges of E' from it. If $E' = \{e\}$ for some edge e of G, then $G - \{e\}$ is abbreviated by $G - e$.

A *matroid* $\mathcal{M} = (S, \mathcal{I})$ is a pair formed by a finite (possibly empty) *ground set* S and a family of subsets $\mathcal{I} \subseteq 2^S$ called *independent sets* satisfying the *independence axioms*:

(I1) $\emptyset \in \mathcal{I}$,

(I2) for any $X, Y \subseteq S$ with $X \subseteq Y$, if $Y \in \mathcal{I}$, then $X \in \mathcal{I}$,

(I3) for any $X, Y \in \mathcal{I}$ with $|X| < |Y|$, there exists $e \in Y - X$ such that $X \cup \{e\} \in \mathcal{I}$.

The maximal independent subsets of S are called *bases*. The *rank* of a set $X \subseteq S$ in the matroid, denoted by $r(X)$, is the maximum size of an independent subset of X. The *rank of the matroid* is the

rank of its ground set. If $\mathcal{M}$ is a matroid on the ground set S and $T \subseteq S$, then the *restriction* of $\mathcal{M}$ to T, or in other words, the *deletion* of $S - T$ from $\mathcal{M}$, is the matroid $\mathcal{M}\big|_T := (T, \mathcal{I}')$, or also denoted by $\mathcal{M} \setminus (S - T) := (T, \mathcal{I}')$, where $\mathcal{I}' := \{X \subseteq T \mid X \in \mathcal{I}\}$. Furthermore, if the rank of $\mathcal{M}\big|_T$ equals that of $\mathcal{M}$, i.e. $r(T) = r(S)$, then we say that $\mathcal{M}\big|_T$ is a *rank-preserving restriction* of $\mathcal{M}$. A *coloring* of a matroid is an arbitrary assignment of colors to the elements of its ground set. A *graphic matroid* is a matroid whose independent sets can be represented as the edge sets of forests of a graph.

Since the number of independent sets can be exponential in the size of the ground set, the usual requirement for a matroid algorithm to be polynomial is to be polynomial in the size of the ground set and not in the size of the input matroid. For this, it is assumed that the input matroid is given by an *oracle* – in our case, by an *independence oracle*, and with an independence oracle call we can determine whether a subset of the ground set is independent in the matroid – and when analyzing the complexity of the matroid algorithm, the oracle calls are counted as single steps.

Definition 1 *We say that a (not necessarily properly) edge-colored graph G is* edge-color-avoiding connected *if after the removal of the edges of any single color from G, the remaining graph is connected.*

For two small examples on the definition of edge-color-avoiding connectivity, see Figure 1.

Figure 1: An example for an edge-color-avoiding connected graph (left) – after the removal of edges of any single color, there remains a Hamiltonian path –, and an example for a not edge-color-avoiding connected graph (right) – after the removal of the blue (denoted by squares) edges, the bottom right vertex becomes isolated.

Clearly, an edge-colored graph is edge-color-avoiding connected if and only if after the removal of the edges of any single color, there exists a spanning tree in the remaining graph. This motivates the introduction of the following definition for matroids, which we call, after DeVos et al. [1], courteously colored matroids.

Definition 2 *Let $\mathcal{M}$ be a matroid, whose ground set is colored. We say that $\mathcal{M}$ is a courteously colored matroid if after the deletion of the elements of any single color from $\mathcal{M}$, the rank of the matroid does not change.*

Thus a matroid is courteously colored if and only if after the deletion of the elements of any single color from the ground set, at least one basis remains intact. In particular, a graphic matroid is courteously colored if and only if each component of the corresponding graph is edge-color-avoiding connected. Another simple example is the case of uniform matroids. The ground set of a uniform matroid $U_{n,k}$ is of size n, and its independent sets are those subsets of the ground set whose cardinality is at most k for some integer $0 \leq k \leq n$. We note that $U_{n,k}$ is graphic if and only if $k \in \{0, 1, n-1, n\}$. It is not difficult to see that the uniform matroid $U_{n,k}$ is courteously colored if and only if there exists no color which is assigned to at least $n - k + 1$ elements.

For convenience, let us introduce the following notation.

Notation 3 *Given an edge-colored graph G and a color c, we denote by $G_{\overline{c}}$ the graph which can be obtained from G by removing the edges of color c from it.*

Given a matroid $\mathcal{M}$ whose ground set is colored and given a color c, we denote by $\mathcal{M}_{\overline{c}}$ the matroid which can be obtained from $\mathcal{M}$ by deleting the elements of color c from it.

3 Courteously colored rank-preserving restrictions of a matroid

In this section, we study the problem of finding edge-color-avoiding connected spanning subgraphs with minimum number of edges in edge-color-avoiding connected graphs, or in general, finding courteously colored rank-preserving restrictions to a set of minimum size in courteously colored matroids.

It is not difficult to see that an edge-colored graph G has an edge-color-avoiding connected spanning subgraph if and only if G is edge-color-avoiding connected, and similarly, a colored matroid $\mathcal{M}$ has a courteously colored rank-preserving restriction if and only if $\mathcal{M}$ is courteously colored.

Theorem 4 *Given a matroid $\mathcal{M} = (S, \mathcal{I})$ whose ground set is colored and given a positive integer m, it is NP-complete to decide whether $\mathcal{M}$ has a courteously colored rank-preserving restriction to a subset $T \subseteq S$ of size at most m.*

Furthermore, this problem remains NP-complete even for graphic matroids.

Proof: The problem is clearly in NP. Now we show that the problem is NP-hard even for graphic matroids. As we observed earlier, a graphic matroid is courteously colored if and only if each component of the corresponding graph is edge-color-avoiding connected. In addition, note that if every edge has a different color, then a component of the graph is edge-color-avoiding connected if and only if it is 2-edge-connected. Thus for those connected graphs in which every edge has a different color and for the choice of $m = \big|V(G)\big|$, our problem is equivalent to deciding whether the graph contains a Hamiltonian cycle, which is known to be NP-complete [3]. □

Corollary 5 *Given an edge-colored graph $G = (V, E)$ and a positive integer m, it is NP-complete to decide whether G has an edge-color-avoiding connected spanning subgraph with at most m edges.*

As is clear from the proof of Theorem 4, in the case of connected graphs whose edges are all of different colors, we want to find a 2-edge-connected spanning subgraph with minimum number of edges, which is an NP-hard problem. However, there exists approximation algorithms for this latter problem. Khuller and Vishkin [8] provided a 3/2-approximation algorithm for it: they modified the depth-first search algorithm so that it does not just find a spanning tree, but a minimally 2-edge-connected[1] spanning subgraph. Gabow et al. [2] presented a $\big(1 + \frac{2}{k}\big)$-approximation algorithm for finding a k-edge-connected spanning subgraph with minimum number of edges with the use of linear programming. Currently, the best known approximation factor for finding a 2-edge-connected spanning subgraph with minimum number of edges is $\frac{4}{3}$ by Hunkenschröder et al. [6], but there exist better performing algorithms if the input graph satisfies some additional conditions – for example, see [11, 15].

In the following, we present a polynomial-time approximation algorithm for finding a courteously colored rank-preserving restriction of a matroid to a set of minimum size. To shorten the description of the algorithm, let us define the following subroutine.

Subroutine IncreaseRank

Input: a matroid $\mathcal{M} = (S, \mathcal{I})$ and a subset $T \subseteq S$.
Output: a set $T' \subseteq S$ for which $r(T') = r(S)$ and $T \subseteq T'$.

```
T' ← T
for s ∈ S do
    if r(T') < r(T' ∪ {s}) then
        T' ← T' ∪ {s}
return T'
```

Now we are ready to present the algorithm.

[1] A graph is called minimally k-edge-connected if it is k-edge-connected but after the removal of any of its edges, the obtained graph is not k-edge-connected.

Algorithm 1 Finding courteously colored rank-preserving restrictions

Input: a courteously colored matroid $\mathcal{M} = (S, \mathcal{I})$ with $S \neq \emptyset$, colored with a color set C.
Output: a courteously colored rank-preserving restriction of $\mathcal{M}$.

```
T ← a basis of M
for c ∈ C do
    T_c̄ ← {s ∈ T | s is not of color c}
    if r(T_c̄) < r(S) then
        T ← T ∪ IncreaseRank(M_c̄, T_c̄)
for s ∈ T do
    if (M|_T) \ {s} is a courteously colored rank-preserving restriction of M then
        T ← T − {s}
return M|_T
```

Remark 6 *Note that if the rank of the input matroid $\mathcal{M}$ is zero, then for any color c, the rank of $\mathcal{M}_{\overline{c}}$ is also zero and the only basis of $\mathcal{M}$ is the empty set. Thus the algorithm selects the empty set at the first step and does not add any elements to it later, so the output is $T = \emptyset$. This is clearly the optimal solution.*

To analyze the case when the rank of the input matroid is at least one, first we prove the following theorem.

Theorem 7 *Let $\mathcal{M} = (S, \mathcal{I})$ be a courteously colored matroid, colored with exactly $k \in \mathbb{Z}_+$ colors, and let $r = r(S)$. If $r \geq 1$, then $k \geq 2$ and $|S| \geq \left\lceil \frac{k \cdot r}{k-1} \right\rceil$, and these lower bounds are tight.*

Proof: Suppose to the contrary that there exists a courteously colored matroid $\mathcal{M}$ colored with $k = 1$ color c, and with rank $r \geq 1$. Then by the definition of courteous colorings, $\mathcal{M}_{\overline{c}}$ has rank r. On the other hand, the ground set of $\mathcal{M}_{\overline{c}}$ is the empty set, which is a contradiction.

Now consider the case $k \geq 2$. Let $\mathcal{M}$ be a courteously colored matroid with rank r, where the elements of the ground set are colored with exactly k colors. Then for any of these k colors c, the matroid $\mathcal{M}_{\overline{c}}$ has rank r, thus its ground set has at least r elements. That sums up to at least $k \cdot r$ elements, where every element is counted exactly $k - 1$ times, so the number of elements is at least $\left\lceil \frac{k \cdot r}{k-1} \right\rceil$.

To show that these lower bounds are tight, we construct courteously colored graphic matroids of rank r on $\left\lceil \frac{k \cdot r}{k-1} \right\rceil$ elements which are colored with exactly k colors for any $k \geq 2$. More precisely, we construct an edge-color-avoiding connected graph $G = (V, E)$ on $r + 1$ vertices and with $r + \left\lceil \frac{r}{k-1} \right\rceil = \left\lceil \frac{k \cdot r}{k-1} \right\rceil$ edges, where the edges are colored with exactly k colors. Let

$$C := \{0, 1, \ldots, k-1\}$$

be the color set, let

$$V := \{v_0, \ldots, v_r\}$$

with $r \geq k - 1$, and let

$$E_i := \left\{ v_j v_{j+1} \;\middle|\; j \in \{0, 1, \ldots, r-1\} \text{ and } j \equiv i \pmod{k-1} \right\}$$

be the set of edges of color i for any $i \in \{0, 1, \ldots, k-2\}$, and let

$$E_{k-1} := \left\{ v_j v_{\max(j+k-1, r)} \;\middle|\; j \in \{0, \ldots, r-1\} \text{ and } j \equiv 0 \pmod{k-1} \right\}$$

be the set of edges of color $k - 1$. Note that G might have a pair of parallel edges between the vertices v_{r-1} and v_r; for an example see Figure 2.

It is not difficult to show that G is edge-color-avoiding connected and has $r + \left\lceil \frac{r}{k-1} \right\rceil = \left\lceil \frac{k \cdot r}{k-1} \right\rceil$ edges. □

In the following theorem, we analyze Algorithm 1.

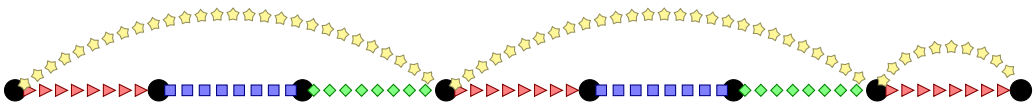


Figure 2: An edge-color-avoiding connected graph on 8 vertices and with minimum number of edges colored with exactly 4 colors.

Theorem 8 *Algorithm 1 is a polynomial-time $\frac{2(k-1)}{k}$-approximation algorithm for finding a courteously colored rank-preserving restriction of a courteously colored matroid – given by an independence oracle – whose elements are colored with exactly $k \in \mathbb{Z}_+$ colors to a set of minimum size.*

Moreover, there exist inputs for which the approximation ratio is exactly $\frac{2(k-1)}{k}$.

Proof: Let $\mathcal{M} = (S, \mathcal{I})$ be the input matroid whose ground set is courteously colored with exactly $k \in \mathbb{Z}_+$ colors and let $r := r(S)$. If $k = 1$, then by Theorem 7, $r = 0$ must hold, and by Remark 6, Algorithm 1 finds an optimal solution in this case.

Now assume $k \geq 2$. In the first step, we simply select a basis of $\mathcal{M}$. Note that this step already guarantees the rank-preserving property of the output.

Now we show that in the second phase, the algorithm selects some additional elements of the ground set to ensure that the output is courteously colored. Since $\mathcal{M}$ is courteously colored, $\mathcal{M}_{\overline{c}}$ has the same rank as $\mathcal{M}$, namely r, thus we can select some additional elements from the ground set of $\mathcal{M}_{\overline{c}}$ so that the obtained set of selected elements has rank r as well. Therefore, at the end of the second phase, the restriction of $\mathcal{M}$ to the so far selected elements is indeed courteously colored.

In the third phase, the algorithm deselects some elements while maintaining the desired properties of the output. Therefore, the algorithm finds an appropriate subset of the ground set. Note that the elements of the output are not necessarily colored with exactly k colors.

Now we prove that the ground set of the output contains at most $2\frac{k-1}{k} \cdot \frac{k \cdot r}{k-1} = 2r$ elements, which is, by Theorem 7, at most $2\frac{k-1}{k}$ times as many as the minimum number of elements of a courteously colored matroid whose elements are colored with at most k colors, implying that Algorithm 1 is a $\frac{2k-1}{k}$-approximation algorithm.

In the first step, the algorithm selects a basis B of $\mathcal{M}$, which clearly consists of r elements.

In the second phase, for each color c, if the deletion of the elements of color c decreases the rank of the set of the so far selected elements, then the algorithm selects some additional elements of some colors different from c to avoid this happening. More precisely, if the deletion of the elements of color c decreases the rank of the set of the so far selected elements by x_c, then the algorithm selects x_c new elements of some colors different from c. For any color c, let y_c denote the number of elements of color c in the basis B found in the first step. Since B is a basis, the deletion of y_c elements from B decreases its rank by exactly y_c. However, there might be some additional selected elements of colors different from c, thus the algorithm selects at most y_c elements for every color c in the second phase. Therefore, at the end of the second phase, at most

$$r + \sum_{c \in C} y_c = r + |B| = 2r$$

elements are selected.

In the third phase, the algorithm only deselects some elements, thus the ground set of the output indeed contains at most $2r$ elements.

Next we prove that the algorithm runs in polynomial time if $\mathcal{M}$ is given by an independence oracle.

The first step is selecting a basis which can be done in $O\big(|S|\big)$ time with the greedy algorithm.

In the second phase, for every color $c \in C$, we remove the elements of color c from the set of the so far selected elements – which can be done in $O\big(|S|\big)$ time –, then we select some additional elements with the

use of Subroutine IncreaseRank – which can be done in $O(|S|)$ time as well. Thus the algorithm takes $O(|C| \cdot |S|)$ steps in the second phase.

In the third phase, the algorithm checks for every selected element s whether for any color $c \in C$, the deletion of s and all the selected elements of color c decreases the rank of the set of the so far selected elements. For a given selected element s and a given color c, this can be done in $O(|S|)$ time with the greedy algorithm. Thus the algorithm takes $O(|C| \cdot |S|^2)$ steps in the third phase.

Therefore, the algorithm runs in $O(|C| \cdot |S|^2)$, i.e. in polynomial time.

Finally, we present some courteously colored matroids, more precisely, some edge-color-avoiding connected graphs, for which the approximation ratio is exactly $\frac{2(k-1)}{k}$.

Let us define the edge-colored graph $G = (V, E)$ as follows. Let

$$C := \{0, 1, \ldots, k-1\}$$

be the color set, let

$$V := \{v_0, \ldots, v_{n-1}\}$$

with $k - 1 \mid n - 1$, and let

$$E_i := \left\{ v_j v_{j+1} \;\middle|\; j \in \{0, 1, \ldots, n-2\} \text{ and } j \equiv i \pmod{k-1} \right\}$$

and

$$E'_i := \left\{ v_j v_{j+1} \;\middle|\; j \in \{0, 1, \ldots, n-2\} \text{ and } j + 1 \equiv i \pmod{k-1} \right\}$$

be the sets of edges of color i for any $i \in \{0, 1, \ldots, k-2\}$, and let

$$E_{k-1} := \left\{ v_j v_{j+k-1} \;\middle|\; j \in \{0, \ldots, n-2\} \text{ and } j \equiv 0 \pmod{k-1} \right\}$$

be the set of edges of color $k - 1$. For an example, see Figure 3.

By Theorem 7, the subgraph $(V, \cup_{i \in C} E_i)$ is an optimal solution with $\frac{k \cdot (n-1)}{k-1}$ edges. However, the output of the algorithm can also be the subgraph $\left(V, \cup_{i \in C \setminus \{k-1\}} (E_i \cup E'_i)\right)$. To see this, first note that the edges of both $\cup_{i \in C \setminus \{k-1\}} E_i$ and of $\cup_{i \in C \setminus \{k-1\}} E'_i$ form Hamiltonian paths $v_0 v_1 \ldots v_{n-1}$, which are disjoint with each other. Thus the algorithm can select the edges of $\cup_{i \in C \setminus \{k-1\}} E_i$ in the first step (these edges obviously form a spanning tree), then it can select all the edges of $\cup_{i \in C \setminus \{k-1\}} E'_i$ in the second phase (all the so far selected edges clearly form an edge-color-avoiding connected graph), and then it does not deselect any edges in the third phase. Clearly, this output has $2n - 1$ edges, therefore the approximation ratio in this case is $\frac{2(k-1)}{k}$. □

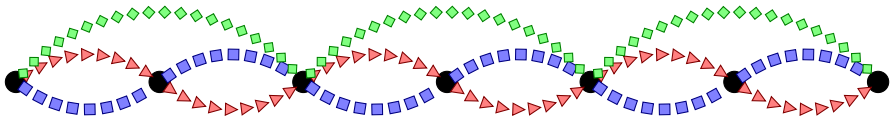


Figure 3: An edge-color-avoiding connected graph colored with $k = 3$ colors and on $n = 7$ vertices, which contains an edge-color-avoiding connected spanning subgraph with $\frac{k \cdot n - 1}{k-1} = 9$ edges – for example, such a subgraph is spanned by the green (denoted by rhombi) edges and the lower red (denoted by triangles) and blue (denoted by squares) edges – and an edge-color-avoiding connected spanning subgraph with $2(n-1) = 12$ edges – that subgraph is spanned by the red and blue edges. The output of Algorithm 1 can be this latter subgraph, resulting in an approximation ratio of exactly $\frac{2(k-1)}{k}$.

As a consequence, we also obtained the following result.

Corollary 9 *Let $\mathcal{M} = (S, \mathcal{I})$ be a courteously colored matroid with rank r such that $\mathcal{M} \setminus \{s\}$ is not courteously colored for all $s \in S$. Then $|S| \leq 2r$ and this upper bound is sharp.*

For a construction of a courteously colored matroid (more precisely, of an edge-color-avoiding connected graph) of rank r, which has maximum number of elements (edges) for the property that none of the elements can be deleted such that the matroid remains courteously colored, see Figure 4.

Figure 4: An edge-color-avoiding connected graph on $n = 8$ vertices and with $2(n-1) = 14$ edges colored with exactly $k = 4$ colors, and having the property that none of the edges can be removed such that the graph remains edge-color-avoiding connected.

Remark 10 *Using Theorem 7 and Corollary 9, one can design a simpler (but for large networks, less efficient) polynomial-time* $\frac{2(k-1)}{k}$*-approximation algorithm for finding courteously colored rank-preserving restrictions of a matroid* $\mathcal{M}$ *to a set of minimum size: one by one greedily delete those elements of the ground set from* $\mathcal{M}$ *after whose deletion the obtained matroid is courteously colored and has the same rank as* $\mathcal{M}$*. By the same reasoning as that for Algorithm 1, we get that there exist inputs for which the approximation ratio of this greedy algorithm is exactly* $\frac{2(k-1)}{k}$*.*

As we observed before, the underlying graph of a courteously colored graphic matroid is not necessarily edge-color-avoiding connected, thus we also present a separate polynomial-time approximation algorithm for finding edge-color-avoiding connected spanning subgraphs with minimum number of edges. (Note that by Corollary 5, this problem is NP-hard.)

To simplify the description of the algorithm, we introduce the following simple subroutines. The subroutine **Graph**(V, E) creates a graph with vertex set V and edge set E, the subroutine **SpanningTree**(G) returns the edges of a spanning tree of a connected graph G, the subroutine **ConnectedComponents**(G) returns the family of the vertex sets of the connected components of G. Finally, the subroutine **ContractVertices**$(G, \mathcal{W})$, whose inputs are a graph G and a partition $\mathcal{W}$ of its vertex set, returns a graph which can be obtained from G by contracting the vertices of each set $W \in \mathcal{W}$ into a single vertex. Finally, the subroutine **BeforeContraction**(G, H, E'), whose inputs are a graph G, and a graph H that is obtained from G by contracting some of its vertices, and an edge set $E' \subseteq E(H)$, returns a set of edges in G corresponding to E'.

Algorithm 2 Finding edge-color-avoiding connected spanning subgraphs

Input: an edge-color-avoiding connected graph $G = (V, E)$ colored with a color set C.
Output: an edge-color-avoiding connected spanning subgraph G' of G.

```
E' ← SpanningTree(G)
G' ← Graph(V, E')
for c ∈ C do
    if G'_c is not connected then
        W ← ConnectedComponents(G'_c)
        H ← ContractVertices(G_c, W)
        E' ← E' ∪ BeforeContraction(G, H, SpanningTree(H))
        G' ← Graph(V, E')
for e ∈ E' do
    if G' − e is edge-color-avoiding connected then
        G' ← G' − e
return G'
```

Analogously to Theorem 4, it can be proved that Algorithm 2 is a $\frac{2(k-1)}{k}$-approximation algorithm of running time $O\big(|C| \cdot |V|^2\big)$ for the problem of finding an edge-color-avoiding connected spanning subgraph

with minimum number of edges of a given graph $G = (V, E)$ colored with a color set C (for a detailed proof, see [16]).

For (not necessarily properly) vertex-colored graphs, we give two definitions of color-avoiding connectivity describing slightly different phenomena. We say that two vertices u and v of a vertex-colored graph are internally vertex-c-avoiding connected for some color c if there exists a u-v path containing no internal vertices of color c. We say that two vertices u and v of a vertex-colored graph are vertex-c-avoiding connected for some color c if they are internally vertex-c-avoiding connected, or there exists a u-v path and at least one of the vertices u and v is of color c. If any two vertices of a vertex-colored graph are vertex- or internally vertex-c-avoiding connected for any color c, then the graph is called vertex- or internally vertex-color-avoiding connected, respectively.

The problems of finding a vertex- or an internally vertex-color-avoiding connected spanning subgraph with minimum number of edges of a given graph are also NP-hard problems by the same reasoning as that for edge-color-avoiding connectivity in Theorem 4. Similar polynomial-time approximation algorithms to Algorithm 2 can be developed for vertex- and internally vertex-color-avoiding connectivity with approximation factors of 2 and slightly better than 3, respectively [16].

Acknowledgement. The authors would like to express their gratitude to Roland Molontay for useful conversations and for his support during the research process.

References

[1] M. DeVos, T. Johnson, and P. Seymour, Cut coloring and circuit covering, `https://web.math.princeton.edu/~pds/papers/cutcolouring/paper.pdf` (2006)

[2] H. N. Gabow, M. X. Goemans, É. Tardos, and D. P. Williamson, Approximating the smallest k-edge connected spanning subgraph by LP-rounding, *Networks* **53**(4), 345–357 (2009)

[3] M. R. Garey and D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, *W. H. Freeman and Company* (1979)

[4] M. Giusfredi and F. Bagnoli, A self-organized criticality method for the study of color-avoiding percolation, *In Proceedings of the Internet Science: 6th International Conference (INSCI 2019)*, 217–226 (2019)

[5] M. Giusfredi and F. Bagnoli, From color-avoiding to color-favored percolation in diluted lattices, *Future Internet* **12**(8), 139 (2020)

[6] C. Hunkenschröder, S. Vempala, and A. Vetta, A 4/3-approximation algorithm for the minimum 2-edge connected subgraph problem, *ACM Transactions on Algorithms* **15**(4), 55 (2019)

[7] A. Kadović, S. M. Krause, G. Caldarelli, and V. Zlatić, Bond and site color-avoiding percolation in scale free networks, *Physical Review E* **98**(6), 062308 (2018)

[8] S. Khuller and U. Vishkin, Biconnectivity approximations and graph carvings, *Journal of the ACM* **41**(2), 214–235 (1994)

[9] S. M. Krause, M. M. Danziger and V. Zlatić, Hidden connectivity in networks with vulnerable classes of nodes, *Physical Review X* **6**(4), 041022 (2016)

[10] S. M. Krause, M. M. Danziger and V. Zlatić, Color-avoiding percolation, *Physical Review E* **96**(2), 022313 (2017)

[11] P. Krysta and V. S. A. Kumar, Approximation algorithms for minimum size 2-connectivity problems, *STACS 2001: Proceedings of the 18th Annual Symposium on Theoretical Aspects of Computer Science*, 431–442 (2001)

[12] L. Lichev, Color-avoiding percolation of random graphs: between the subcritical and the intermediate regime, https://arxiv.org/abs/2301.09910 (2023)

[13] L. Lichev and B. Schapira, Color-avoiding percolation on the Erdős-Rényi random graph, https://arxiv.org/abs/2211.16086 (2022)

[14] R. Molontay and K. Varga, On the complexity of color-avoiding site and bond percolation, *SOFSEM 2019: Theory and Practice of Computer Science*, 354–367 (2019)

[15] V. V. Narayan, A 17/12-approximation algorithm for 2-vertex-connected spanning subgraphs on graphs with minimum degree at least 3, https://arxiv.org/abs/1612.04790 (2016)

[16] J. Pintér, Extremal problems of color-avoiding connectivity, *Master's Thesis, Budapest University of Technology and Economics* (2022)

[17] B. Ráth, K. Varga, P. T. Fekete, and R. Molontay, Color-avoiding percolation in edge-colored Erdős–Rényi graphs, https://arxiv.org/abs/2208.12727 (2022)

Generalized solution for the Herman Protocol Conjecture

Endre Csóka[1]

Alfréd Rényi Institute of Mathematics
csoka.endre@renyi.hu

Szabolcs Mészáros[2]

Department of Mathematics
Central European University
szabolcs.thai@gmail.com

András Pongrácz

Alfréd Rényi Institute of Mathematics
pongracz.andras@renyi.hu

Abstract: Herman's classical self-stabilizing algorithm is a process where an odd number of tokens is placed on some of N fixed positions along a circle. In every step, we move each token to the clockwise neighboring position or leave it in its current position with equal probability, independently for all tokens. Colliding tokens are removed, and the process ends when only one token remains. The Herman Protocol Conjecture states that the expected time $\mathbb{E}(\mathbf{T})$ of Herman's algorithm is at most $\frac{4}{27}N^2$. We prove the conjecture in its standard unbiased and also in a biased form for discrete processes, and extend the result to further variants where the tokens move via certain Lévy processes. Moreover, we derive a bound on the expected value of $\mathbb{E}(\alpha^{\mathbf{T}})$ for all $1 \leq \alpha \leq (1-\varepsilon)^{-1}$ with a specific $\varepsilon > 0$. Subject to the correctness of an optimization result that can be demonstrated empirically, all these estimations attain their maximum on the initial state with three tokens distributed equidistantly on the ring of N processes. Such a relation is the symptom of the fact that both $\mathbb{E}(\mathbf{T})$ and $\mathbb{E}(\alpha^{\mathbf{T}})$ are weighted sums of the probabilities $\mathbb{P}(\mathbf{T} \geq t)$.

Keywords: Stochastic processes, Discrete optimization, Random algorithms, Stochastic optimization.

1 Introduction

The simplified setup of Herman's self-stabilizing algorithm consists of a directed circular graph of N elements and K tokens put on K different nodes of the graph. The vertices represent identical processes connected along the edges. Ideally, if the system is in a legitimate state, only one process holds a token in the configuration. However, errors may occur when the system enters into a multiple token state. Herman's algorithm is a randomized protocol to reach a one-token state after an error, hence the name self-stabilizing; cf. [4, 5].

The method of the algorithm is the following: in every step of the discretely treated time, if a process holds a token then it keeps it with probability $\frac{1}{2}$ or passes it to its clockwise neighbor with probability $\frac{1}{2}$, independently of the other token-passes. If a process kept its token in a step but also receives one, then both tokens disappear. By the implementation of the processes, we can guarantee that Herman's algorithm starts at a configuration where there is an odd number of tokens, hence the mentioned algorithm will eventually yield a one-token state with probability 1. We note that it is just as reasonable from a

[1]Research is supported by the NRDI grant KKP 138270 and by the European Research Council (grant agreement no.˜306493).

[2]Research is supported by the NRDI grant KKP 138270.

mathematical point of view to start with an even number of tokens and run the process until all tokens disappear. Interestingly, this setup is much easier to handle, and was solved in [6].

Several questions arise naturally about the distribution of the execution time of self-stabilization, i.e., the hitting time $\mathbf{T}$ of a one-state configuration [12]. Since the complete description of the distribution $\mathbb{P}(\mathbf{T} \geq t)$ did not turn out to be an accessible problem, the analysis focused mainly on the derived quantity $\mathbb{E}(\mathbf{T})$. The denominator, Herman, proved the upper bound $\frac{1}{2}N^2 \log N$ on $\mathbb{E}(\mathbf{T})$ in the original paper [10], which was improved to $O(N^2)$ by multiple authors independently [6, 8, 13, 14].

To find a tight bound, it is reasonable to search for the extremum of $\mathbb{E}(\mathbf{T})$ as a function of the initial configuration of the tokens. Assuming that the stabilization process starts with three tokens, the maximum of $\mathbb{E}(\mathbf{T})$ is realized on the equidistant starting position of the tokens (or the closest configuration to that, if N is not divisible by 3). This is a consequence of the description of $\mathbb{E}(\mathbf{T})$ given by [13] for all the initial configurations with three tokens. They found an explicit formula for $\mathbb{E}(\mathbf{T})$ in terms of the "distances" of the tokens, where by distance of the tokens X_1 and X_2 we mean the length of the arc connecting X_1 and X_2 avoiding the third token X_3. Given these distances a, b, $c \in \mathbb{N}$ of the tokens (where necessarily $a + b + c = N$ by definition), the expectation of $\mathbf{T}$ can be expressed as

$$\mathbb{E}(\mathbf{T}) = \frac{4abc}{N}$$

This expression clearly has the maximum at the states where a, b, c are the nearest integers around $\frac{N}{3}$ summing up to N. In particular, $a = b = c = \frac{N}{3}$ if N is divisible by 3. In [13], it was also conjectured to be the only maximum of $\mathbb{E}(\mathbf{T})$ considering all possible initial configurations, not necessarily with three tokens.

We give a proof to this conjecture by using a method that can be generalized in many directions. In the original phrasing of the conjecture, in every round of the discrete process, each token either keeps its current position or makes a move in the positive direction, and both events occur with equal probability $p = \frac{1}{2}$. We treat the unbiased version where all tokens make a move independently with the same probability p; cf. [7, 16]. The argument is not purely combinatorial in its intrinsic nature. Hence, it can be generalized to the case when the movement of each token is described by a Poisson process or when the circular graph is replaced by a continuous circle on which the tokens move by independent Brownian motions with the same parameters.

Moreover, we show that in the (unbiased) discrete version of the protocol with parameters p, N, and for $\varepsilon = 4p(1-p)\sin^2\left(\frac{\pi}{2N}\right)$, we have $\mathbb{E}\left(\left(\frac{1}{1-\varepsilon}\right)^{\mathbf{T}}\right) \leq \frac{3}{2}$, with equality if and only if we start from the three-token equidistant configuration. Furthermore, subject to the the correctness of an optimization result that we tested by computer, the maximum of $\mathbb{E}\left(\alpha^{\mathbf{T}}\right)$ is attained at the three-token equidistant configuration for all $1 \leq \alpha \leq \frac{1}{1-\varepsilon}$. All the evidence point towards the potential result that $\mathbb{P}(\mathbf{T} \geq t)$ is maximized by the equidistant three-token configuration for all t. Indeed, $\mathbb{E}\left(\alpha^{\mathbf{T}}\right)$ is also a linear combination of the $\mathbb{P}(\mathbf{T} \geq t)$ just like $\mathbb{E}(\mathbf{T})$, but with weights $\alpha^t - \alpha^{t-1}$ rather than 1's, as in the case of $\mathbb{E}(\mathbf{T})$. In [12], a formula was established to $\mathbb{P}(\mathbf{T} \geq t)$ assuming that there are three tokens. As a consequence, they have shown that the maximum of $\mathbb{P}(\mathbf{T} \geq t)$ is indeed attained at the equidistant three-token starting state when we consider the three-token initial states. The next step could be to obtain this theorem with no restriction on the number of tokens.

This paper is an extended and improved version of the manuscript [3], which proved the Herman Protocol Conjecture parallelly with and independently of [1].

2 Main results

2.1 Basic setup and three-token states

We implement a somewhat modified viewpoint on the described processes. Since the arguments require some symmetry, for our purposes it is better to rotate the base space by $\frac{2\pi}{2N}$ after every step counter-clockwise, where N stands for the number of nodes. This slight notational modification have the effect

that the number of nodes gets doubled, but half of them is necessarily avoided by the tokens in every step. At least this is the case in the classical, discrete version of the protocol, but not necessarily in the other variants we consider in this paper. In the sequel we refer to $2N$ as the number of nodes. Moreover, the tokens now move in a symmetrized way. In the standard Herman protocol, they either move to the clockwise neighboring new node with probability $\frac{1}{2}$ or move in the opposite direction to the counter-clockwise neighbor with probability $\frac{1}{2}$, all independently.

We now generalize this setup by choosing another parameter (besides N).

Definition 1 *Given a $p \in]0,1[$, the discrete Herman protocol with parameters p, N is the process where a number of tokens are moved independently along a circle with $2N$ nodes, such that each token moves to its counter-clockwise neighboring node with probability p, and to the other neighbor with probability $1-p$, all independently. In other words, tokens are taking independent biased random walks* $\pmod{2N}$.

Using this reformulation, it is natural to consider further variants. We define two more setups that we can handle in essentially the same way as the classical discrete version. In these new variants, the movement of each token is a continuous time process, in fact, a Lévy process. It is vital in some arguments that the tokens cannot jump over one another. Hence, if we intend to preserve the symmetry between the tokens in the sense that the processes describing their movements are independent copies of the same Lévy process, only the following two special types can be considered.

Definition 2 *In the exponential clock (or Poisson) variant, the tokens are still positioned on the $2N$ nodes along the circle, and steps are discrete. Each token moves to its neighbor in the positive direction with probability p, and to the other one with probability $1-p$. However, the timeline is continuous, and each token has a corresponding exponential clock with mean 1: whenever it goes off, the token takes a step.*

Definition 3 *The Brownian (or Wiener) variant is continuous, making the nodes irrelevant. The tokens can be positioned anywhere around the circle with perimeter $2N$, and they each independently move via a Brownian motion with variance 1. In this setup, N need not be an integer.*

If one were only interested in the Brownian variant, a somewhat more natural parametrization could be chosen. However, in order to be consistent with the other variants, and to easily compare the three situations, we use this one.

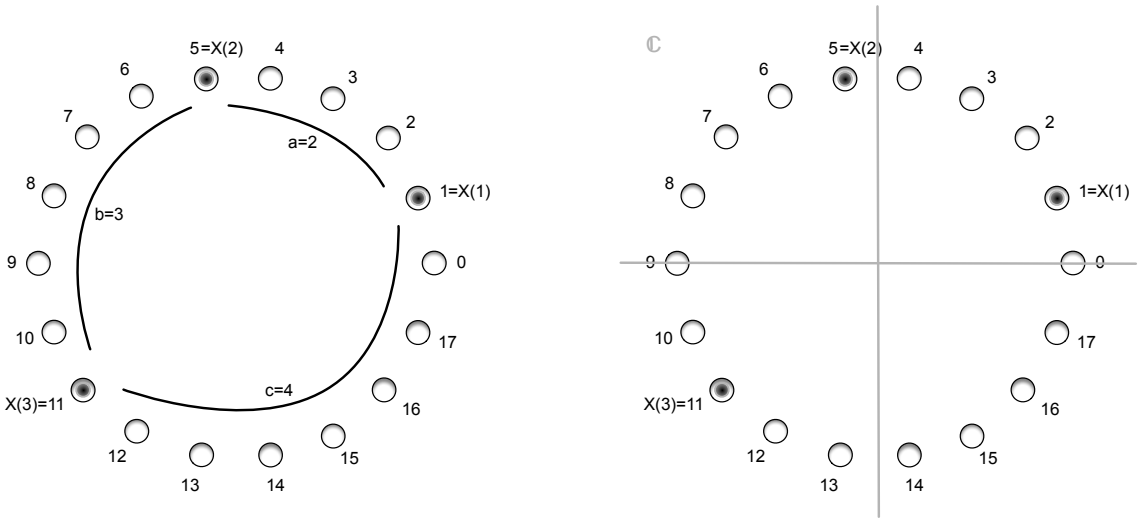


Figure 1: *Illustration of a three-token configuration with $N = 9$. On the left, the distances between the tokens are indicated. On the right, the same configuration is placed in the complex plane.*

The three-token states play a crucial role in the argument, thus we first determine the expected time to absorption from such initial states. The formula in the unbiased discrete setup ($p = \frac{1}{2}$) was shown in [15]. In the standard discrete protocol, a three-token state was usually represented by three numbers a, b, c, the distances between the three pairs of tokens, i.e. the number of nodes on the arcs connecting two tokens while avoiding the third. Then the expected runtime of the process is $\frac{4abc}{N}$ according to [15]. Note that in our modified setup with the circle having perimeter $2N$ rather than N, we require a slight adjustment to get the same formula: the distances a, b, c are now half the arc lengths connecting each pair of tokens, as the nodes are doubled. So in order to obtain $\frac{abc}{27}$ for the expected runtime for three-token states, in our formulation with the $2N$ nodes and each arc between neighboring nodes having length 1, the actual arc lengths between the tokens are $2a$, $2b$, $2c$, respectively. In order to keep the consistency, we still refer to the distances as a, b, c; see the left hand side of Figure 2.1. This is a common feature for all three variants considered in the paper. Throughout the paper, we make several references to the *equidistant three-token configuration* without elaborating on the cases where $2N$ is not divisible by 3.

The following lemma is essentially first-step analysis.

Lemma 4 *Let x be an initial three-token state with distances a, b, c between the three pairs of tokens. Then the Herman protocol is expected to terminate in $\mathbb{E}(\mathbf{T}(x)) = \mu\frac{abc}{N}$ steps, where*

- *$\mu = \frac{1}{p(1-p)}$ for the discrete version with parameters p, N;*
- *$\mu = 4$ for the exponential clock variant with parameters p, N;*
- *$\mu = 4$ for the Brownian process with parameter N.*

In particular, we have the upper bound $\mathbb{E}(\mathbf{T}(x)) \leq \frac{\mu N^2}{27}$ for all three-token states x, with equality if and only x is the equidistant three-token configuration. Moreover, $\mathbb{E}(\mathbf{T})$ is finite for all three variants and arbitrary initial states.

The main goal of the paper is to generalize the upper bound $\mathbb{E}(\mathbf{T}(x)) \leq \mu\frac{N^2}{27}$ to all possible initial states x for all three variants of the protocol.

Theorem 5 *Let $\mathbf{T}(x)$ denote the hitting time of the one-token state starting from the initial state x for any of the three variants of the Herman protocol. Then $\mathbb{E}(\mathbf{T}(x)) \leq \frac{\mu N^2}{27}$, with equality if and only if x is the equidistant three-token configuration, where $\mu = \frac{1}{p(1-p)}$ for the discrete variant, and $\mu = 4$ for the exponential clock version and for the Brownian process.*

To verify the bound in Theorem 5, it seems natural not only to keep count on when the process is terminated, but to have a way to measure "how far" we are from the end in expectation. Then the goal becomes to show that this measure is the worst (i.e. the highest) throughout the whole process if and only if the initial state is the equidistant three-token configuration. To this end, we define two "potentials" that are expected to grow, and that start off and end up in $[0, 1]$. The first such potential is Φ, see Definition 6. It is only relevant for three-token states; it assigns to a three-token state x the expected value of the remaining time until the process terminates starting from x, rescaled into $[0, 1]$. Note that Lemma 4 guarantees that Φ is indeed between 0 and 1.

Definition 6 *Let x be an initial three-token state with distances a, b, c between the three pairs of tokens. Then for all three variants of the Herman process we define $\Phi(x) = 1 - \mathbb{E}(\mathbf{T}(x))/\left(\mu\frac{N^2}{27}\right) = 1 - \frac{27abc}{N^3}$.*

2.2 The other potential

The following arguments are explained for the discrete variant of the process, but it is easy to see that everything generalizes to the other versions, as well. The only relevant nontrivial property that we make use of is that the tokens have a circular order that cannot change without a collision of tokens. That is,

tokens cannot jump through one another without first occupying the same position; this was exactly the property that we focused on when defining the three variants.

To define the non-trivial potential, the core idea of the paper, we number the nodes by 0, 1, 2, ..., $2N-1$. The location of the j-th token at time $t \in \mathbb{N}_{\geq 0}$ is described by the random variable $X_t(j)$ where $j = 1, 2, \ldots, K_t$ and K_t stands for the number of tokens at time t, where the tokens are numbered compatible to their ordering on the circle (but the beginning of the enumeration is arbitrary). Generalizing the notation K_t we will write $K_t(x)$ for the (random) number of tokens at time t for the process starting at the initial state x. In particular, $K(x) := K_0(x)$ denotes the number of tokens at state x. As before, $\mathbf{T} := \mathbf{T}(x) := \min\{t \mid K_t(x) = 1\}$ is the hitting time of a one-token state, i.e. the execution time of the self-stabilizing algorithm. Note that this notion is not affected by the symmetrization of the process we implemented in the previous chapter. Also, we need a notation for the hitting time of a three-token state, as it turns out to be a crucial point in the evolution of the process, so we put $\tau := \min\{t \mid K_t(x) = 3\}$.

Just like $\Phi(x)$, the new potential $x \mapsto \Psi(x) \in [0,1]$ also measures how far our state is from the final state in expectation. The growth speed of Ψ can be estimated without trying to compute the first potential Φ for all configurations with an arbitrary number of tokens, a seemingly impossible challenge. However, we can show that the two potentials are reasonably close to each other on three-token states; see Lemma 11.

In the definition of Ψ, we use the complex exponential function $k \mapsto e^{\frac{2\pi i}{2N}k} = e^{\frac{\pi i}{N}k}$. This notation also implicitly contains an identification of the circle with the complex unit circle (the identification was essentially chosen when we numbered the nodes); see the right side of Figure 2.1. This arbitrary choice could in principle cause some trouble. But as we will soon see, the potential Ψ is invariant under rotation of the circle (i.e., the choice of the node with number 0), solving the issue; see Proposition 8.

Definition 7 *Let x be an arbitrary state and assume that $0 \leq x(1) < x(2) < \cdots < x(K) \leq 2N-1$ where $x(j)$ is the position of the j'th token of the state x using counter-clockwise enumeration of the tokens starting at the direction $1 \in \mathbb{C}$ (the node with number 0); see the right side of Figure 2.1. Then the potential Ψ is*

$$\Psi(x) := \Big| \sum_{j=1}^{K(x)} e^{\frac{\pi i}{N}\frac{1}{2}x(j)}(-1)^j \Big|^2$$

Geometrically, $x \mapsto \Psi(x)$ can be described as summing up the (directed) angle bisectors of the vectors $e^{\frac{\pi i}{N}x(j)}$ and the fixed unit vector 1 with an extra twist. Namely, for odd j we reflect the resulting angle bisector vector to the origin. Informally, this reflection is applied to stabilize the quantity under the disappearance of two colliding tokens. Formally, it means that if $x(j) = x(j+1)$ then deleting these two tokens from the vector x does not change the value of $\Psi(x)$. Also the alternating sign is responsible for the independence of Ψ from the choice of the direction, i.e., the identification of the circle with the unit circle in the complex plane.

Proposition 8 *For any state x we have*

1. *The choice made at the identification of the plane with $\mathbb{C}$ does not affect $\Psi(x)$. That is, $\Psi(x)$ is invariant under the simultaneous translation of the $x(j)$, even if during the translation, a token jumps over $1 \in \mathbb{C}$.*
2. *The disappearance of two colliding tokens does not affect $\Psi(x)$.*
3. *$\Psi(x) \leq 1$, with equality if and only if there is only one token at state x.*
4. *$\Psi(x) \geq 0$, with equality if and only if x is an equidistant configuration with at least three tokens.*

Now, we fix the initial state x of the process $t \mapsto X_t$. Let us denote by $Y_t = \Psi(X_t)$ the value of the potential defined above on the random process at time t. As usual, $\mathcal{F}_t := \sigma(X_t(j) \mid s \leq t,\ j \leq K_0)$ is the standard filtration of the process, and the number of tokens $K_t = K_t(x)$ at time t was defined earlier.

The evolution of Y_t is described by the following lemma.

Lemma 9

- *Given the discrete version of the Herman protocol with parameters p, N, let $\varepsilon = 4p(1-p)\sin^2\left(\frac{\pi}{2N}\right)$. Then for any $t \in \mathbb{N}$ we have*
$$\mathbb{E}(Y_{t+1} - Y_t \mid \mathcal{F}_t) = \varepsilon(K_t - Y_t)$$
- *In the exponential clock variant, let $\varepsilon = 4\sin^2\left(\frac{\pi}{4N}\right)$. Then $\lim_{\Delta t \to 0} \frac{1}{\Delta t}\mathbb{E}(Y_{t+\Delta t} - Y_t \mid \mathcal{F}_t) = \varepsilon(K_t - Y_t)$.*
- *In the Brownian version, let $\varepsilon = \frac{\pi^2}{4N^2}$. Then $\lim_{\Delta t \to 0} \frac{1}{\Delta t}\mathbb{E}(Y_{t+\Delta t} - Y_t \mid \mathcal{F}_t) = \varepsilon(K_t - Y_t)$.*

Corollary 10 *For all three versions of the Herman protocol we have $\mathbb{E}(Y_\tau) \geq 4\varepsilon\mathbb{E}(\tau) + Y_0$.*

Proof: Observe that Lemma 9, item 3. of Proposition 8, and the fact that in a state with more than three tokens there are at least five tokens imply

$$\mathbb{E}(Y_{t+\Delta t} - Y_t|\mathcal{F}_t) \geq \varepsilon\Delta t(5-1) = 4\varepsilon\Delta t$$

whenever $t < \tau$. Here, $\Delta t = 1$ for the discrete variant and an infinitesimally small amount of time in the other two versions. Hence, the process

$$Z_t = \begin{cases} Y_t - 4\varepsilon t & \text{if } t < \tau \\ Y_\tau - 4\varepsilon\tau & \text{if } t \geq \tau \end{cases}$$

is a submartingale. To show that Z_t is indeed integrable, first note that $|Y_t| \leq 1$ yields the trivial bound $|Z_t| \leq 1 + 4\varepsilon t$. In particular, if $t/\mathbb{E}(\tau) \leq 1$ then $\mathbb{E}|Z_t| \leq 1 + 4\varepsilon \cdot \mathbb{E}(\tau)$; note that $\mathbb{E}(\tau)$ is finite according to Lemma 4. If $t/\mathbb{E}(\tau) = C > 1$ then we can apply the Law of Total Expectation and the Markov inequality to obtain

$$\mathbb{E}|Z_t| = \mathbb{P}(t < \tau) \cdot \mathbb{E}\left(|Z_t| \mid t < \tau\right) + \mathbb{P}(t \geq \tau) \cdot \mathbb{E}\left(|Z_t| \mid t \geq \tau\right) \leq \frac{1}{C}(1 + 4\varepsilon C \cdot \mathbb{E}(\tau)) + 1 \cdot \mathbb{E}|Y_\tau - 4\varepsilon\tau| \leq$$

$$\leq \frac{1}{C}(1 + 4\varepsilon C \cdot \mathbb{E}(\tau)) + \mathbb{E}(1 + 4\varepsilon\tau) \leq \frac{1}{C} + 4\varepsilon \cdot \mathbb{E}(\tau) + 1 + 4\varepsilon \cdot \mathbb{E}(\tau) \leq 2 + 8\varepsilon \cdot \mathbb{E}(\tau)$$

Hence, according to the Optional Stopping Theorem $\mathbb{E}(Z\tau) \geq \mathbb{E}(Z_0)$, and consequently, $\mathbb{E}(Y_\tau) - 4\varepsilon\mathbb{E}(\tau) \geq Y_0$.

□

Informally, this corollary means that the potential Ψ grows fast enough until we hit a three-token state. After time τ however, it slows down, as we can only guarantee a $2\varepsilon \cdot \Delta t$ growth under a (small) time period of length Δt by the same argument as in the proof of Corollary 10. Fortunately, we have an exact formula to the other potential Φ for three-token states, see Definition 6. As we will see, it is easy to find an exact formula for Ψ on three-token states, as well. So the vague idea is to estimate the growth of the potential Ψ before the hitting time τ of three-token states by Corollary 10, and then switch to the other potential Φ. In order to show that this switch can be carried out without a major loss in expectation, we need to compare the two potentials on three-token states.

Lemma 11 *For any state x with three tokens $\Phi(x) \geq 0.87 \cdot \Psi(x)$.*

We are ready to prove the main result of the paper.

Proof: *(of Theorem 5)* First, let's investigate what happens to the potential Φ at the moment of the potential interchange:

$$\Phi(X_\tau) = 1 - \frac{\mathbb{E}(\mathbf{T}(x))|_{x=X_\tau}}{\mu\frac{N^2}{27}} = 1 - \frac{27}{\mu N^2}\mathbb{E}\big(\mathbf{T}(X_\tau) \,|\, X_\tau\big) = 1 - \frac{27}{\mu N^2}\mathbb{E}\big(\mathbf{T} - \tau \,|\, X_\tau\big)$$

Hence, taking expectation yields

$$\mathbb{E}\big(\Phi(X_\tau)\big) = 1 - \frac{27}{\mu N^2}\mathbb{E}(\mathbf{T} - \tau)$$

So now, we can estimate $\mathbb{E}(\mathbf{T})$ as:

$$\mathbb{E}(\mathbf{T}) = \mathbb{E}(\tau) + \mathbb{E}(\mathbf{T} - \tau) = \mathbb{E}(\tau) + \frac{\mu N^2}{27} \cdot \Big(1 - \mathbb{E}\big(\Phi(X_\tau)\big)\Big) \leq$$

where we can apply Lemma 11:

$$\leq \mathbb{E}(\tau) + \frac{\mu N^2}{27} \cdot \Big(1 - 0.87 \cdot \mathbb{E}\big(\Psi(X_\tau)\big)\Big) = \mathbb{E}(\tau) + \frac{\mu N^2}{27} \cdot \Big(1 - 0.87 \cdot \mathbb{E}(Y_\tau)\Big) \leq$$

So we can use the estimation of Y_τ proved in Corollary 10:

$$\leq \mathbb{E}(\tau) + \frac{\mu N^2}{27} \cdot \Big(1 - 0.87 \cdot \big(4\varepsilon\mathbb{E}(\tau) + Y_0\big)\Big) = \frac{\mu N^2}{27} + \left(1 - \frac{1.16}{9}\mu\varepsilon N^2\right)\mathbb{E}(\tau) - \frac{0.29}{9}\mu N^2 Y_0$$

To finish the proof, it suffices to show that the second term is negative, or equivalently, $1 - \frac{1.16}{9}\mu\varepsilon N^2 < 0$. In the discrete version $\mu\varepsilon = 4\sin^2\left(\frac{\pi}{2N}\right)$, in the Poisson variant $\mu\varepsilon = 16\sin^2\left(\frac{\pi}{4N}\right)$, and in the Brownian version $\mu\varepsilon = \frac{\pi^2}{N^2}$. Thus $\mu\varepsilon N^2$ is roughly $\pi^2 \approx 9.87$ in all three cases. To be more accurate, this is exactly the case for the Brownian version. In the other two variants, the worst constant is obtained for $N = 3$: $4\sin^2\left(\frac{\pi}{2\cdot 3}\right) \cdot 3^2 = 9$ and $16\sin^2\left(\frac{\pi}{4\cdot 3}\right) \cdot 3^2 \approx 9.65$, making the coefficient $1 - \frac{1.16}{9}\mu\varepsilon N^2$ negative. To see the case of equality, note that in the last inequality we estimated from below $\mathbb{E}(\tau)$ by zero and Y_0 by zero as well. If we did not lose anything here, then $\mathbb{E}(\tau) = 0$, thus we start from a three-token state. Moreover, $Y_0 = 0$ hence we started from the equidistant configuration by item 4. of Proposition 8. □

3 More general estimates

We note that the last proof provides a somewhat stronger statement than the conjecture in all three variants. In the right hand side of the inequality $\mathbb{E}(\mathbf{T}) \leq \frac{\mu N^2}{27} + \left(1 - \frac{1.16}{9}\mu\varepsilon N^2\right)\mathbb{E}(\tau) - \frac{0.29}{9}\mu N^2 Y_0$, the coefficient $1 - \frac{1.16}{9}\mu\varepsilon N^2$ is at most -0.23 (assuming that $N \geq 5$, as otherwise $\mathbb{E}(\tau) = 0$), so we obtain $\mathbb{E}(\mathbf{T}) + 0.23\mathbb{E}(\tau) \leq \frac{\mu N^2}{27} - \frac{0.29}{9}\mu N^2 Y_0$. In fact, Corollary 10 can be slightly improved by noticing that the speed of the expected elevation of Ψ is at least $2k\varepsilon\Delta t$ in states with $K = 2\ell - 1$ tokens. Putting $\tau_k := \min\{t \mid K_t(x) = 2\ell - 1\}$, this yields the refined estimate

$$\mathbb{E}(\mathbf{T}) + 0.23\mathbb{E}(\tau) + \sum_{\ell=3}^{(\ell_0+1)/2} 0.61\mathbb{E}(\tau_\ell) \leq \frac{\mu N^2}{27} - \frac{0.29}{9}\mu N^2 Y_0$$

Informally, this means that even if we reward the process for being in states with more than three tokens by making the contribution of such a step a linear function of ℓ (roughly 0.61ℓ rather than 1, as in the computation of the runtime $\mathbf{T}$), the maximum of the expected total contribution is still attained at the three-token equilibrium state.

We now focus on the discrete variant of the process, and show that the presented method can yield further estimates to the distribution of the runtime. We can view the process as an absorbing Markov chain; cf. [11] for an introduction. As usual, the transition matrix is given in a canonical form: that is, indices corresponding to absorbing states (those with one token) are at the end, making the transition matrix a block matrix of the form $P = \begin{pmatrix} Q & R \\ 0 & I \end{pmatrix}$. Moreover, if the states are clustered according to

the number of tokens in them, then Q is also a block matrix, all of whose diagonal blocks are non-negative irreducible matrices. The spectral radius ϱ of Q carries an important probabilistic meaning: clearly, the supremum of those $\alpha \geq 1$ such that $\mathbb{E}(\alpha^{\mathbf{T}})$ is finite is ϱ^{-1}. Moreover, the vector $\underline{u'}$ of values $\mathbb{E}(\alpha^{\mathbf{T}})$ assigned to all non-absorbing states is the restriction of the unique solution to the system of linear equations $\begin{pmatrix} \alpha Q & \alpha R \\ 0 & I \end{pmatrix} \underline{u} = \underline{u}$, where the coordinates of $\underline{u}$ corresponding to absorbing states are all 1. Equivalently, $\begin{pmatrix} Q & R \\ 0 & I \end{pmatrix} \underline{u} = \underline{v}$, where components of $\underline{v}$ corresponding to absorbing states are still 1, and the rest is filled with $1/\alpha$ times the values $\mathbb{E}(\alpha^{\mathbf{T}})$, that is $\underline{v'} = (1/\alpha)\underline{u'}$. As we are interested in such expected values, we compute the spectral radius ϱ of Q, and provide a formula to $\mathbb{E}(\alpha^{\mathbf{T}})$ for three-token initial states. By using the above and the min-max Collatz-Wielandt formula [2], one can verify the following formula.

Lemma 12 *Given the discrete version of the protocol with parameters p, N and $\varepsilon = 4p(1-p)\sin^2\left(\frac{\pi}{2N}\right)$. Then the spectral radius of Q is $\varrho = 1 - 4p(1-p)\sin^2\left(\frac{\pi}{N}\right)$. In particular, $\varrho \approx 1 - 4\varepsilon$, and we have the precise bounds $1 - 4\varepsilon \leq \varrho \leq 1 - 3\varepsilon$. Moreover, given an $\alpha \geq 1$, the expected value $\mathbb{E}(\alpha^{\mathbf{T}})$ is finite if and only if $\alpha < \varrho^{-1}$, and then for three-token states with distances a, b, c between the tokens it is $\mathbb{E}(\alpha^{\mathbf{T}}) = \frac{\beta^a - \beta^{N-a} + \beta^b - \beta^{N-b} + \beta^c - \beta^{N-c}}{1-\beta^N}$, where β is any of the two solutions of the equation $\beta + \beta^{-1} = \frac{\alpha^{-1}-1}{p(1-p)} + 2$.*

Using a slightly rephrased form of Lemma 9 yields the following lemma.

Lemma 13 *Given the discrete version of the protocol with parameters p, N and $\varepsilon = 4p(1-p)\sin^2\left(\frac{\pi}{2N}\right)$. Let $1 \leq \alpha \leq (1-\varepsilon)^{-1}$. Then*

- $\varepsilon\alpha\mathbb{E}\left(\frac{\alpha^{\tau}-1}{\alpha-1}(5 - Y_\tau)\right) \leq \mathbb{E}(Y_\tau) - Y_0$, *and*
- $2\varepsilon\alpha\mathbb{E}\left(\frac{\alpha^{\mathbf{T}}-1}{\alpha-1}\right) \leq 1 - Y_0$.

We note that the second item provides a quadratic upper bound for $\mathbb{E}(\mathbf{T})$ by putting $\alpha \to 1$. Indeed, the left hand side converges to $2\varepsilon\mathbb{E}(\mathbf{T})$ as $\alpha \to 1$, thus $2\varepsilon\mathbb{E}(\mathbf{T}) \leq 1$, and consequently $\mathbb{E}(\mathbf{T}) \leq \frac{1}{2\varepsilon} \approx \frac{2N^2}{\pi^2} \approx 0.203N^2$. This bound is never tight, as we demonstrated in Theorem 5: the tight bound is $\frac{4}{27}N^2 \approx 0.148N^2$. This is the reason we had to cut the process in two: we first estimate the parameters until a three-token state is reached, and then use the precise formulas to the parameters for three-token initial states. However, for one particular choice of α, the second item of the above lemma can yield a tight bound.

Corollary 14 *Given the discrete version of the protocol with parameters p, N and $\varepsilon = 4p(1-p)\sin^2\left(\frac{\pi}{2N}\right)$, we have*

$$\mathbb{E}\left(\left(\frac{1}{1-\varepsilon}\right)^{\mathbf{T}}\right) \leq \frac{3}{2}$$

with equality if and only if we start from the equidistant three-token configuration.

Note that this statement provides a tight bound to a linear combination of the $\mathbb{P}(\mathbf{T} \geq t)$'s with the weights $(1-\varepsilon)^{-t} - (1-\varepsilon)^{-(t-1)} = \varepsilon(1-\varepsilon)^{-t}$.

Proof: By applying the second item of Lemma 13 for $\alpha = (1-\varepsilon)^{-1}$, we have

$$1 \geq 1 - Y_0 \geq 2\varepsilon(1-\varepsilon)^{-1}\mathbb{E}\left(\frac{(1-\varepsilon)^{-\mathbf{T}} - 1}{(1-\varepsilon)^{-1} - 1}\right) = 2\varepsilon\mathbb{E}\left(\frac{(1-\varepsilon)^{-\mathbf{T}} - 1}{1 - (1-\varepsilon)}\right) = 2\mathbb{E}\left((1-\varepsilon)^{-\mathbf{T}} - 1\right)$$

The case of equality holds exactly if we did not lose anything in the estimations. Those in Lemma 13 that were used in the second item are tight if and only if $K_0 = 3$. (Note that the constant $(1-\varepsilon)\alpha - 1 = 0$ if $\alpha = (1-\varepsilon)^{-1}$.) Equality in $1 \geq 1 - Y_0$ holds if and only if $Y_0 = 0$, which is equivalent to the assumption that the tokens are distributed equidistantly by item 4. of Lemma 8. $\square$

Corollary 15 *Given the discrete version of the protocol with parameters p, N and $\varepsilon = 4p(1-p)\sin^2\left(\frac{\pi}{2N}\right)$, we have*

$$\mathbb{E}\left(\frac{5-Y_\tau}{(1-\varepsilon)^\tau}\right) \leq 5$$

with equality if and only if we start from the equidistant three-token configuration.

PROOF: By applying the first item of Lemma 13 for $\alpha = (1-\varepsilon)^{-1}$, we have

$$\mathbb{E}(Y_\tau) \geq \mathbb{E}(Y_\tau) - Y_0 \geq \varepsilon(1-\varepsilon)^{-1}\mathbb{E}\left(\frac{(1-\varepsilon)^{-\tau}-1}{(1-\varepsilon)^{-1}-1}(5-Y_\tau)\right) = \varepsilon(1-\varepsilon)^{-1}\mathbb{E}\left(\frac{(1-\varepsilon)^{-\tau}-1}{\varepsilon(1-\varepsilon)^{-1}}(5-Y_\tau)\right) =$$

$$\mathbb{E}\left(((1-\varepsilon)^{-\tau}-1)(5-Y_\tau)\right) = \mathbb{E}\left(\frac{5-Y_\tau}{(1-\varepsilon)^\tau}\right) - 5 + \mathbb{E}(Y_\tau)$$

□

Theorem 16 *Given the discrete version of the protocol with parameters p, N and $\varepsilon = 4p(1-p)\sin^2\left(\frac{\pi}{2N}\right)$. Let $1 \leq \alpha < (1-\varepsilon)^{-1}$, and let $\gamma = -\log_{1-\varepsilon}\alpha$. For a three-token state x, let $g(x)$ be the expected value of $\alpha^{\mathbf{T}}$ with initial position x; cf. Lemma 12 for the precise formula.*

Then $\sup_{K(x)=3} \frac{g(x)}{(1-\Psi(x)/5)^\gamma}$ is an upper estimate for $\mathbb{E}\left(\alpha^{\mathbf{T}}\right)$ with arbitrary initial state. In particular, if the function $\frac{g(x)}{(1-\Psi(x)/5)^\gamma}$ defined on all three-token states attains its maximum at the three-token equidistant state, then so does $\mathbb{E}\left(\alpha^{\mathbf{T}}\right)$.

We remark that by plotting the function $\frac{g(x)}{(1-\Psi(x)/5)^\gamma}$ it seems evident that the maximum is indeed attained at the three-token equidistant state. However, we do not have a rigorous verification similar to the proof Lemma 11, due to the fact that the current function is more complicated to analyze.

PROOF: Given any initial state, consider the probability distribution of the set of pairs $\{(x,t) \mid K(x) = 3, t \in \mathbb{N}_0\}$ induced by the process: namely, the probability that $t = \tau$ and $x_\tau = x$, that is, we hit the set of three-token states at time t and in the particular state x. Then

$$\mathbb{E}\left(\alpha^{\mathbf{T}}\right) = \mathbb{E}(\alpha^t g(x)) \leq \left(\sup_{K(x)=3}\frac{g(x)}{(5-\Psi(x))^\gamma}\right)\cdot\mathbb{E}\left(\alpha^t(5-\Psi(x))^\gamma\right) = \left(\sup_{K(x)=3}\frac{g(x)}{(5-\Psi(x))^\gamma}\right)\cdot\mathbb{E}\left(\left(\frac{5-\Psi(x)}{(1-\varepsilon)^t}\right)^\gamma\right)$$

By using Jensen's inequality here, and then later Corollary 15, we obtain

$$\mathbb{E}\left(\alpha^{\mathbf{T}}\right) \leq \left(\sup_{K(x)=3}\frac{g(x)}{(5-\Psi(x))^\gamma}\right)\cdot\mathbb{E}\left(\frac{5-\Psi(x)}{(1-\varepsilon)^t}\right)^\gamma \leq \left(\sup_{K(x)=3}\frac{g(x)}{(5-\Psi(x))^\gamma}\right)\cdot 5^\gamma = \sup_{K(x)=3}\frac{g(x)}{(1-\Psi(x)/5)^\gamma}$$

As for the second assertion of the theorem, if the maximum is attained at the three-token equidistant state x_0, then in that state we have $\Psi(x_0) = 0$, thus $\frac{g(x_0)}{(1-\Psi(x_0)/5)^\gamma} = g(x_0)$ is exactly the expected value of $\mathbb{E}\left(\alpha^{\mathbf{T}}\right)$ with the three-token equidistant state as the initial state. □

Another natural problem is to study $\mathbb{E}(\mathbf{T})$ for initial states where there is a token in every original position, i.e., the essentially unique equidistant N-token state (for odd N). Surprisingly, it is useful to combine the two completely different methods in the present paper and in [1].

Proposition 17 *Given an odd integer $N \geq 3$ and $p = 1/2$. Let $\mathbf{T}$ be the runtime of the (unbiased) discrete version of the process from the equidistant N-token state. Then for large enough N we have $CN^2 < \mathbb{E}(\mathbf{T})$, where $C \approx 0.072$.*

References

[1] Maria Bruna, Radu Grigore, Stefan Kiefer, Joël Ouaknine, and James Worrell. Proving the Herman-Protocol Conjecture. In Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi, editors, *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016)*, volume 55 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 104:1–104:12, Dagstuhl, Germany, 2016. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[2] Lothar Collatz. Einschließungssatz für die charakteristischen zahlen von matrizen. *Math. Z.*, 48:221–226, 1942.

[3] Endre Csóka and Szabolcs Mészáros. Generalized solution for the Herman Protocol Conjecture, 2015. https://arxiv.org/pdf/1504.06963v3.pdf.

[4] Edsger W. Dijkstra. Self-stabilizing systems in spite of distributed control. *Commun. ACM*, 17(11):643–644, nov 1974.

[5] Shlomi Dolev. *Self-Stabilization.* MIT Press, 2000.

[6] Yuan Feng and Lijun Zhang. A nearly optimal upper bound for the self-stabilization time in Herman's algorithm. *Distrib. Comput.*, 28(4):233–244, 2015.

[7] Michael E. Fisher. Walks, walls, wetting, and melting. *J. Stat. Phys.*, 34(5–6):667–729, March 1984.

[8] Laurent Fribourg, Stéphane Messika, and Claudine Picaronny. Coupling and self-stabilization. *Distrib. Comput.*, 18(3):221–232, 2006.

[9] Matthias Hammer, Marcel Ortgiese, and Florian Völlering. Entrance laws for annihilating Brownian motions and the continuous-space voter model. *Stochastic Process. Appl.*, 134:240–264, 2021.

[10] Ted Herman. Probabilistic self-stabilization. *Inf. Process. Lett.*, 35(2):63–67, 1990.

[11] John G Kemeny and James Laurie Snell. *Finite Markov chains.* Springer-Verlag, 1976.

[12] Stefan Kiefer, Andrzej S. Murawski, Joël Ouaknine, Björn Wachter, and James Worrell. Three tokens in Herman's algorithm. *Formal Aspects Comput.*, 24(4-6):671–678, 2012.

[13] Annabelle McIver and Carroll Morgan. An elementary proof that Herman's Ring is $\theta(n^2)$. *Inf. Process. Lett.*, 94(2):79–84, 2005.

[14] Annabelle McIver and Carroll Morgan. On the expected time for Herman's probabilistic self-stabilizing algorithm. *Theoret. Comput. Sci.*, 349(3):475–483, 2005.

[15] Andrzej S. Murawski and Joël Ouaknine. On probabilistic program equivalence and refinement. In *CONCUR*, volume 3653 of *Lecture Notes in Computer Science*, pages 156–170. Springer, 2005.

[16] Joachim Rambeau and Grégory Schehr. Distribution of the time at which n vicious walkers reach their maximal height. *Phys. Rev. E, Statistical, nonlinear, and soft matter physics*, 83:061146, 06 2011.

Genericity and maps of matroids

Andras Recski[1]

Department of Computer Science and
Information Theory
Faculty of Electrical Engineering and
Informatics
Budapest University of Technology and
Economics
Műegyetem rkp 3, H-1111 Budapest, Hungary
recski@cs.bme.hu

Abstract: Maps of matroids have been studied for half a century. If two matroids are related by a rank-preserving weak map then their representability properties are often similar. Motivated by electric engineering applications, relations of these questions to the genericity of the entries of the representing matrices are studied.

Keywords: matroid, weak map, strong map, genericity, electric networks

1 Introduction

If a physical system is described by linear conditions then the column space matroid of the coefficient matrix contains many qualitative information about the system (like unique solvability of electric networks, or infinitesimal rigidity of bar and joint frameworks). If a condition is relaxed, the new matroid will be less free than the old one.

Example 1. An infinitesimally rigid planar framework becomes a mechanism if a rod is removed, see Figure 1. The original framework was described by five equations of form $(x_j - x_k)(u_j - u_k) + (y_j - y_k)(i_j - i_k) = 0$ where x_j and y_j denote the coordinates of joint j and u_j and i_j denote their respective time derivatives (this is quite an unusual notation but its reason will become clear soon) where j, k refer to two adjacent joints. The column space matroid of this 5×8 matrix on the underlying set $\{u_1, u_2, u_3, u_4, i_1, i_2, i_3, i_4\}$ has rank five and every 5-tuple is independent except those containing $\{u_1, u_2, u_3, u_4\}$ or $\{i_1, i_2, i_3, i_4\}$. If we remove the rod between joints 1 and 3, the column space matroid of the resulting 4×8 matrix will be less free: four further 4-tuples will become dependent, namely $\{u_1, u_2, i_1, i_2\}, \{u_2, u_3, i_2, i_3\}, \{u_3, u_4, i_3, i_4\}$ and $\{u_4, u_1, i_4, i_1\}$. The 3-dimensional affine representation of this latter matroid is shown on Figure 2 while, in case of the first matroid, the sets $\{u_1, u_2, u_3, u_4\}$ and $\{i_1, i_2, i_3, i_4\}$ are coplanar but otherwise in generic positions within the respective planes, while these planes are in generic position in the 4-dimensional affine space.

Example 2. Consider a 2-port given by the equations $u_1 = R_{11}i_1 + R_{12}i_2$ and $u_2 = R_{21}i_1 + R_{22}i_2$. If both ports are terminated by current sources then the resulting network is uniquely solvable for any values of the R_{jk} parameters. However, if the ports are terminated by voltage sources then the network is uniquely solvable if and only if $R_{11}R_{22} - R_{12}R_{21} \neq 0$. The rank of the 2×4 coefficient matrix does not change if $R_{11}R_{22} - R_{12}R_{21} = 0$ but the column space matroid of the matrix changes from the rank 2 uniform matroid to the one where the set $\{i_1, i_2\}$ is dependent.

[1]The research reported in this paper and carried out at the Budapest University of Technology and Economics was supported by the TKP2020, National Challenges Program of the National Research Development and Innovation Office (BME NC TKP2020) and by the Higher Education Excellence Program of the Ministry of Human Capacities in the frame of the Artificial Intelligence research area of the Budapest University of Technology and Economics (BME FIKP-MI/SC). Useful remarks of Dávid Szeszlér and Áron Vékássy are also greatly appreciated.

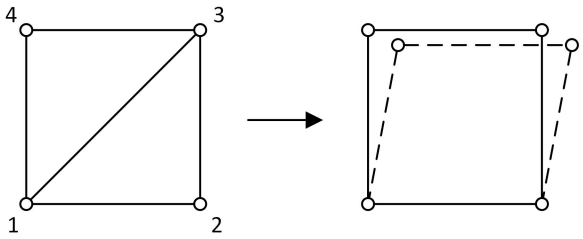


Figure 1: The two planar frameworks of Example 1

Observe that the less free matroid has the same rank as the other in Example 2 and it has lower rank in Example 1. Recall that two matroids $\mathcal{A}, \mathcal{B}$ on the same underlying set S are in strong map relation $\mathcal{A} \Rightarrow \mathcal{B}$ if every closed set of $\mathcal{B}$ is closed in $\mathcal{A}$ as well and they are in weak map relation $\mathcal{A} \to \mathcal{B}$ if every independent set of $\mathcal{B}$ is independent in $\mathcal{A}$ as well. The two matroids of Example 1 are in strong (and also in weak) map relation while those of Example 2 are in weak map relation only.

The following properties of these maps are well known:

(P1) The strong map relation implies the weak one.

(P2) If a matrix $\mathbf{A}$ represents a matroid $\mathcal{A}$ over a field F and the matrix $\mathbf{B} = \mathbf{TA}$ represents the matroid $\mathcal{B}$ over F, where $\mathbf{T}$ is any matrix of appropriate size over F, then $\mathcal{A} \Rightarrow \mathcal{B}$.

(P3) If $\mathcal{A} \neq \mathcal{B}$ then $\mathcal{A} \Rightarrow \mathcal{B}$ implies that the rank of $\mathcal{B}$ is smaller than the rank of $\mathcal{A}$ but $\mathcal{A} \to \mathcal{B}$ may hold between matroids of the same rank as well.

(P4) If $\mathcal{A} \neq \mathcal{B}$ then $\mathcal{A} \Rightarrow \mathcal{B}$ implies $\mathcal{B}^* \Rightarrow \mathcal{A}^*$ for their duals. On the other hand, $\mathcal{A} \to \mathcal{B}$ implies $\mathcal{A}^* \to \mathcal{B}^*$ in case of rank preserving weak maps while $\mathcal{A}^*$ and $\mathcal{B}^*$ can be incomparable if the rank of $\mathcal{B}$ is smaller than the rank of $\mathcal{A}$.

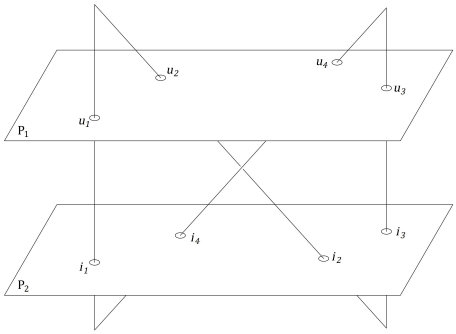


Figure 2: The affine representation of the second matroid of Example 1

2 An example for rank preserving weak maps

In order to illustrate the rich variety of maps consider a 2×4 matrix of rank 2. Due to the applications in electric network theory (see below) let its columns be denoted by u_1, u_2, i_1, i_2. Without loss of generality we may suppose that the first two columns are linearly independent like in Example 2, that is, let the matrix be $\begin{pmatrix} -1 & 0 & a & b \\ 0 & -1 & c & d \end{pmatrix}$. The four quantities a, b, c, d may have "generic" values (say, they are algebraically independent transcendentals over the field of the rationals), then the column space matroid of the matrix is the uniform matroid of rank 2. This is the freest possible case (Case 1 in Table 1 below). If they are not algebraically independent then, among the infinitely many possible algebraic relations

among these four numbers, there are 16 further cases leading to distinct matroids, see the following table, where the meaning of columns $3 \ldots 8$ will be explained later.

case	algebraic relation	No. of bases	*	rec	ISD	φ^*	ESD
1	none	6	1	-	+	1	+
2	$a = 0$	5	5	?		1	+
3	$b = 0$	5	4	-		4	
4	$c = 0$	5	3	-		3	
5	$d = 0$	5	2	?		5	+
6	$ad - bc = 0$	5		?		6	+
7	$a = d = 0$	4	7	?	+	7	+
8	$b = c = 0$	4	8	+	+	8	+
9	$a = b = 0$	3		-		11	
10	$c = d = 0$	3		-		12	
11	$a = c = 0$	3		-		9	
12	$b = d = 0$	3		-		10	
13	$a = b = c = 0$	2		+		13	+
14	$a = b = d = 0$	2		-		15	
15	$a = c = d = 0$	2		-		14	
16	$b = c = d = 0$	2		+		16	+
17	$a = b = c = d = 0$	1		+		17	+

Table 1

The Hasse diagram of Figure 3 shows the set of these 17 matroids ordered by the rank preserving weak map relation. The number of the bases in these matroids, as shown in Column 3 of Table 1, helps to visualize this diagram.

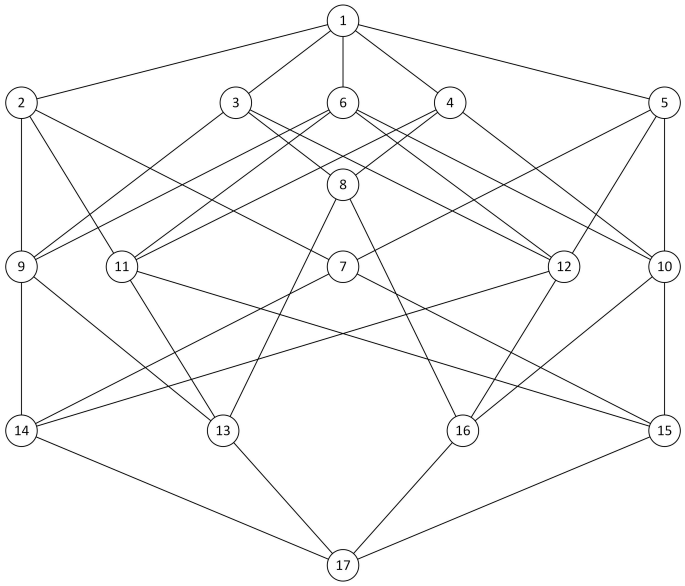


Figure 3: The relations among the 17 matroids given in Table 1

All these matroids except the one on the top are graphic. This phenomenon illustrates an old theorem of Lucas[7]: If $\mathcal{A} \to \mathcal{B}$ is a rank-preserving weak map and $\mathcal{A}$ is graphic then so is $\mathcal{B}$.

3 Relation to electric network theory

A system of linear equations of form $\mathbf{Au}+\mathbf{Bi}=\mathbf{0}$ describes an abstract object called n-port, like in Example 2. The columns of the matrix $\mathbf{M}=(\mathbf{A}\mid\mathbf{B})$ are denoted by $u_1,u_2,u_3,\ldots,u_n,i_1,i_2,i_3,\ldots,i_n$ and can be interpreted as the voltages and the currents of the respective ports of a physical device. Let the number of the rows of $\mathbf{M}$ be k and we may suppose that these rows are linearly independent, that is, $r(\mathbf{M})=k$. An n-port is ordinary if $k=n$. If $\mathbf{C}$ is a nonsingular $k\times k$ matrix then $(\mathbf{A}\mid\mathbf{B})$ and $(\mathbf{CA}\mid\mathbf{CB})$ describe the same n-port, hence the two matrices are called equivalent and denoted by $(\mathbf{A}\mid\mathbf{B})\approx(\mathbf{CA}\mid\mathbf{CB})$.

Let two n-ports be defined by $\mathbf{A}_1\mathbf{u}+\mathbf{B}_1\mathbf{i}=\mathbf{0}$ and by $\mathbf{A}_2\mathbf{u}+\mathbf{B}_2\mathbf{i}=\mathbf{0}$, respectively. They are called the negative of each other if $(\mathbf{A}_1\mid\mathbf{B}_1)\approx(\mathbf{A}_2\mid-\mathbf{B}_2)$ and the inverse of each other if $(\mathbf{A}_1\mid\mathbf{B}_1)\approx(\mathbf{B}_2\mid\mathbf{A}_2)$. They are called the dual of each other if $r(\mathbf{A}_1\mid\mathbf{B}_1)+r(\mathbf{A}_2\mid\mathbf{B}_2)=2n$, and $\mathbf{u}_1^T\mathbf{u}_2+\mathbf{i}_1^T\mathbf{i}_2=0$ holds for any pairs $(\mathbf{u}_1,\mathbf{i}_1)$ and $(\mathbf{u}_2,\mathbf{i}_2)$ satisfying $\mathbf{A}_1\mathbf{u}_1+\mathbf{B}_1\mathbf{i}_1=\mathbf{0}$ and $\mathbf{A}_2\mathbf{u}_2+\mathbf{B}_2\mathbf{i}_2=\mathbf{0}$ (where T denotes transpose).

If $\mathbf{M}=(\mathbf{A}\mid\mathbf{B})$ is a matrix description of an n-port then its column space matroid on the set $S=\{u_1,u_2,u_3,\ldots,u_n,i_1,i_2,i_3,\ldots,i_n\}$ is denoted by $\mathcal{M}(\mathbf{M})$. Observe that equivalent n-ports have identical matroids and the matroids of dual n-ports (if duality is defined as above) are dual to each other. Hence every ordinary n-port has a dual. However, Table 1 contains only those 2-ports where $\{u_1,u_2\}$ is independent, hence their dual appears in seven cases only; see Column 4 of the table.

An n-port is reciprocal if its dual equals to its negative inverse. If $\mathbf{A}=\mathbf{E}$ then an n-port given by $\mathbf{M}=(\mathbf{A}\mid\mathbf{B})$ is reciprocal if and only if $\mathbf{B}$ is symmetric. Clearly, this condition is usually not reflected by the column space matroid of $\mathbf{M}$, hence reciprocity cannot always be determined from the zero-nonzero pattern of $\mathbf{B}$. Column 5 of Table 1 indicates if the 2-ports of the respective cases are

(1) always (+) reciprocal (since $b=c$),

(2) sometimes (?) reciprocal, or

(3) never (–) reciprocal (since either $b=0$ and $c\neq 0$, or $c=0$ and $b\neq 0$, or they have generic values).

Recall that a matroid $\mathcal{M}$ with the underlying set S is called self-dual if there exists a permutation π of S satisfying $\pi(\mathcal{M})=\mathcal{M}^*$. If this permutation is the identity then the matroid is called identically self-dual (ISD). Column 6 of Table 1 indicates that three of the 17 matroids are ISD.

Let φ denote the permutation of the set $S=\{u_1,u_2,u_3,\ldots,u_n,i_1,i_2,i_3,\ldots,i_n\}$ satisfying $\varphi(u_k)=i_k$ and $\varphi(i_k)=u_k$ for every k. If two n-ports $\mathbf{M}_1$ and $\mathbf{M}_2$ are the inverse of each other then $\varphi(\mathcal{M}_1)=\mathcal{M}_2$. Hence, we obtain:

Statement: If an n-port given by $\mathbf{M}=(\mathbf{A}\mid\mathbf{B})$ is reciprocal then its matroid $\mathcal{M}$ satisfies $\varphi(\mathcal{M})=\mathcal{M}^*$.

However, $\varphi(\mathcal{M})=\mathcal{M}^*$ does not guarantee the reciprocity of the n-port. Column 7 of Table 1 shows the effect of taking the permutation φ and duality thereafter: There are 9 cases satisfying $\varphi(\mathcal{M})=\mathcal{M}^*$ but only 4 of them guarantees reciprocity.

Matroids satisfying $\varphi(\mathcal{M})=\mathcal{M}^*$ are called electrically self-dual (ESD), see [10]. Column 8 of Table 1 shows that eight of the 17 matroids are ESD. n-ports with ESD matroids are called qualitatively reciprocal, they have applications in electric network synthesis [9].

The recently introduced concept of potentially reciprocal multiports [11] justify the study of the strong maps of ESD-matroids.

4 A problem on representability

The examples of Table 1 suggest that if $\mathcal{A}\to\mathcal{B}$ is a rank-preserving weak map and $\mathcal{A}$ is represented by a matrix $\mathbf{A}$ over a field then the representation of $\mathcal{B}$ can be obtained by imposing some extra algebraic relations among the nonzero entries of $\mathbf{A}$. However, this is not true in general. For example, $\mathcal{F}^-\to\mathcal{F}$ satisfies the above condition (where $\mathcal{F}^-$ and $\mathcal{F}$ denote the anti-Fano and the Fano matroid, respectively), but they cannot be represented over a common field.

On the other hand, we conjecture a slightly weaker statement:

Conjecture Let $\mathcal{A} \to \mathcal{B}$ be a rank-preserving weak map between two different matroids and suppose that both are representable over the field of the reals. Then there exists a matrix $\mathbf{A}$ representing $\mathcal{A}$ so that a representation of $\mathcal{B}$ can be obtained by imposing some extra algebraic relations among the nonzero entries of $\mathbf{A}$.

In fact, we present an algorithm and conjecture that it always gives the right construction. At first we illustrate the idea by an example referring to the seven matroids shown in Figure 4. Each matroid is given by its affine representation (left) and by its graphic representation, if it exists (right).

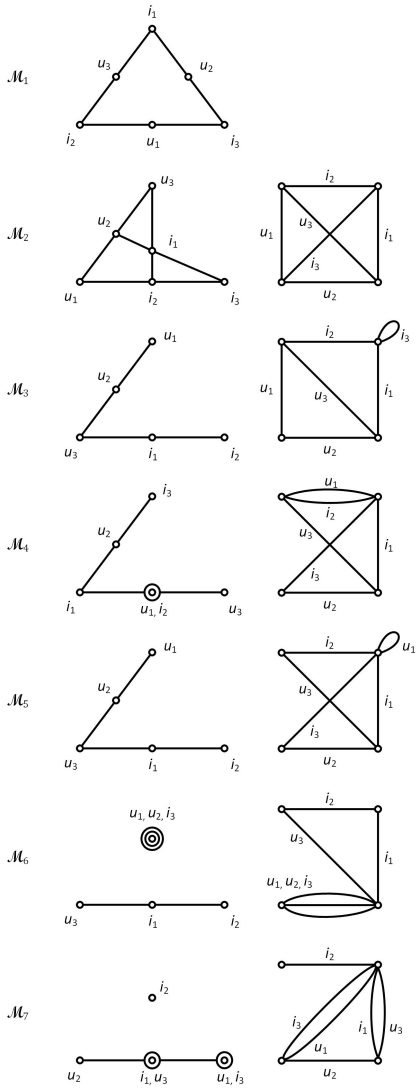


Figure 4: The matroids of the example

Most people would represent matroid $\mathcal{M}_2$ over the field of the reals by the matrix

$$\begin{pmatrix} 0 & -1 & 1 & 1 & 0 & 0 \\ 1 & 0 & -1 & 0 & 1 & 0 \\ -1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

(where the columns are denoted by $u_1, u_2, u_3, i_1, i_2, i_3$, respectively), but then changing a nonzero entry to zero need not necessarily lead to a weak map. For example, if we change the last or the first nonzero

entry of the third row to zero, we obtain the matroids $\mathcal{M}_3$ and $\mathcal{M}_4$, respectively. One can easily see that $\mathcal{M}_2 \to \mathcal{M}_3$ holds but $\mathcal{M}_2 \to \mathcal{M}_4$ does not hold. If we replace all the nonzero entries of the first three columns of the above matrix to generic values then the obtained matrix

$$\begin{pmatrix} 0 & c & e & 1 & 0 & 0 \\ a & 0 & f & 0 & 1 & 0 \\ b & d & 0 & 0 & 0 & 1 \end{pmatrix}$$

will represent $\mathcal{M}_1$ rather than $\mathcal{M}_2$, since the relation $ade + bcf = 0$ is not valid for generic values. If we wish to find weak maps of $\mathcal{M}_2$ with $a = 0$, then b, c or f must also be set to 0, leading to $\mathcal{M}_5$, $\mathcal{M}_6$ and to $\mathcal{M}_7$, respectively (if the other two values are kept generic). The Hasse diagram of Figure 5 shows the set of these 7 matroids ordered by the rank preserving weak map relation.

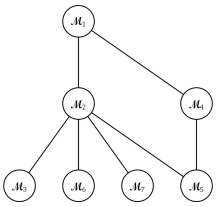


Figure 5: Relations among the seven matroids of the example

This example shows that if $\mathcal{A} \to \mathcal{B}$ is a rank-preserving weak map (say, $\mathcal{M}_2 \to \mathcal{M}_3$) then the entries of the matrix $\mathbf{A}$ representing $\mathcal{A}$ should be set at first to "as close to generic as possible".

Algorithm

Step 1. Let B be a base of $\mathcal{B}$ (and hence a base of $\mathcal{A}$ as well) and let $k = |B|$ be the common rank of the two matroids. Let $\mathbf{A}'$ be a matrix representing $\mathcal{A}$. We may suppose, without loss of generality, that $\mathbf{A}'$ is of form $(\mathbf{E}|\mathbf{A}'')$ where $\mathbf{E}$ is the unity matrix and their columns correspond to the elements of B. Replace all the nonzero elements of $\mathbf{A}''$ by "generic" variables, that is, by real numbers which are algebraically independent over the field of the rationals. This matrix $\mathbf{T}_B(\mathbf{A})$ represents a (fundamental transversal) matroid $\mathcal{T}_B(\mathcal{A})$. Clearly, this is at least as free as $\mathcal{A}$, that is, $\mathcal{T}_B(\mathcal{A}) \to \mathcal{A}$.

Step 2. For each $\ell \leq k$ consider every ℓ-element subset X of the underlying set S of the matroids. If X is independent in $\mathcal{A}$ or if it is dependent in $\mathcal{T}_B(\mathcal{A})$ then do nothing. Otherwise consider the $k \times \ell$ submatrix of $\mathbf{T}_B(\mathbf{A})$ determined by the columns of X. Since these columns are linearly independent, this submatrix has some nonsingular $\ell \times \ell$ submatrices $\mathbf{D}$. For each such submatrix consider the algebraic equation $\det \mathbf{D} = 0$. Let $C[\mathcal{T}_B(\mathcal{A}) \to \mathcal{A}]$ denote the collection of all these equations (obtained for every choice of X). Clearly, if all these equations are met by the variables appearing as entries of $\mathbf{T}_B(\mathbf{A})$ then the obtained matrix $\mathbf{A}$ represents $\mathcal{A}$.

Step 3. Similarly construct the collection $C[\mathcal{A} \to \mathcal{B}]$ of equations. If all these equations are met by the variables appearing as entries of $\mathbf{A}$ then the obtained matrix $\mathbf{B}_0$ represents a matroid $\mathcal{B}_0$.

Clearly, $\mathcal{B} \to \mathcal{B}_0$. The example $\mathcal{A} = \mathcal{U}_{7,3}$ and $\mathcal{B} = \mathcal{F}$ shows that equality need not necessarily hold. But so far we applied the representability of $\mathcal{A}$ only. We conjecture that if $\mathcal{B}$ is also representable over the field of the reals then $\mathcal{B} = \mathcal{B}_0$.

Obviously, this algorithm for finding these algebraic relations is not efficient, since the number of nonsingular $\ell \times \ell$ submatrices to be considered may be superpolynomial.

References

[1] T. Brylawski, Constructions, in N. White (Ed.) *Theory of Matroids*, Cambridge University Press, Cambridge (1986) 127-223

[2] J. Geelen, J. Oxley, D. Vertigan and G. Whittle, Weak Maps and Stabilizers of Classes of Matroids, *Adv. Appl. Math.* **21** (1998) 305-241

[3] M. Iri and A. Recski, What Does Duality Really Mean? *Int. J. Circuit Theory Appl.* **8** (1980) 317-324

[4] M. Iri and A. Recski, Duality and reciprocity – a qualitative approach, Proc. IEEE Internat. Symp. Circuits and Systems, Rome (1982) 415-418

[5] J. P. S. Kung, Strong Maps, in N. White (Ed.) *Theory of Matroids*, Cambridge University Press, Cambridge (1986) 224-253

[6] J. P. S. Kung and H. Q. Nguyen, Weak Maps, in N. White (Ed.) *Theory of Matroids*, Cambridge University Press, Cambridge (1986) 254-271

[7] D. Lucas, Properties of Rank Preserving Weak Maps, *Bull. Amer. Math. Soc.* **80** (1974) 127-131

[8] J. Oxley and G. Whittle, On Weak Maps of Ternary Matroids, *European J. Combinatorics* **19** (1998) 377-389

[9] A. Recski, Matroids and Network Synthesis, Proc. European Conf. on Circuit Theory and Design, Warsaw (1980) 192-197

[10] A. Recski, Some Problems of Self-Dual Matroids, *Coll. Math. Soc. János Bolyai* **37** (1981) 635-648

[11] A. Recski and Á. Vékássy, Interconnection, Reciprocity and a Hierarchical Classification of Generalized Multiports, *IEEE Trans. Circuits Syst, I – Regular Papers* **68** (2021) 3682-3692

Optimal cutting arrangements in 1D

Bowen Li[1]

Carleton College
lib2@carleton.edu

Attila Sali[2]

Alfréd Rényi Institute of Mathematics
and Department of Computer Science
Budapest University of Technology and
Economics
sali.attila@renyi.hu

Abstract: Mathematical model of an industrial application is investigated. Orders with given tolerances must be fit on a warehouse of steel rods so that the number of cuts needed to satisfy the orders is minimized. It is shown that the problem of feasibility and if the orders can be satisfied, then finding the minimum number of cuts needed are both NP-complete. Two practical solution methods are introduced: one is based on dynamic programming and maximum clique search in graphs, the other one uses $0-1$-linear programming. Simulations show that the latter one is much more effective.

NP-completeness, Dynamic programming, MaxClique, $0-1$-linear programming

1 Introduction

In the present paper the following industrial problem [**?**], introduced in the framework of a Slovenian–Hungarian applied mathematics joint project, is treated. We are given a warehouse of steel rods of (maybe) different lengths. Orders of pieces of rods come in and they need to be served. However, cutting the rods is costly, so the number of cuts needs to be minimized. The way to do that is finding *exact fits*, that is collection of orders that fit on some rod of the warehouse and also exhaust that rod totally. In each exact fit case one cut can be saved, as "the remaining piece of the rod" need not be cut off.

The input is given as the status of the *warehouse*, i.e., the lengths of the available steel rods, which is a multiset:

$$W = \{w_1, w_2, \ldots, w_n\},$$

furthermore the *orders* are given as pairs (a, b), also forming a multiset:

$$N = \{(a_1, b_1), \ldots, (a_m, b_m)\}.$$

The pair (a, b) means that some tolerance is allowed, that is we can cut a rod of any length in the interval $[a, b]$. We need to find a partition $\mathcal{P} = \{P_1, P_2, \ldots, P_k\}$ of the multiset N and find a *valid* cutting assignment

$$\pi : \mathcal{P} \to W$$

such that: $\forall P_i \in \mathcal{P} : \sum_{(a,b)\in P_i} a \leq \pi(P_i)$. That is, the the orders that belong to the partition class P_i are assigned to be cut from rod $\pi(P_i)$. Amongst these valid cutting assignments we look for one that minimizes the number of cuts needed. Define the *exact fit* for $w \in W$ be the case when $\pi(P_i) = w$ and $\sum_{(a,b)\in P_i} a \leq \pi(P_i) \leq \sum_{(a,b)\in P_i} b$.

[1]Research was done in Undergraduate Research Course at Budapest Semesters In Mathematics

[2]Research is partially supported by the National Research, Development and Innovation Office (NKFIH) grants K–116769 and SNN–135643. This work was also supported by the BME- Artificial Intelligence FIKP grant of EMMI (BME FIKP-MI/SC) and by the Ministry of Innovation and Technology and the National Research, Development and Innovation Office within the Artificial Intelligence National Laboratory of Hungary.

Fact 1 *If a valid cutting assignment exists then the number of cuts is minimized iff the number of exact fits are maximized.*

However, if a valid cutting assignment is infeasible, then it has to be decided what are the priorities, such as the largest possible number of orders satisfied and within that the minimum number of cuts, or having the minimum number of cuts so that the remaining pieces of rods do not allow any unsatisfied order to be satisfied. These two are clearly different problems, for example if the warehouse is $W = \{1\}$ and the multiset of orders is $\{(0.3, 0.31), (0.3, 0.31), (0.3, 0.31), (0.49, 0, 51), (0.49, 0, 51)\}$, then we can maximize the number of satisfied orders with three cuts, on the other hand using one cut we can exhaust the whole warehouse. Having this in mind we tacitly assume that the inputs allow valid cutting assignments.

2 Theoretical results

In this section we treat the complexity of minimizing the number of cuts. Let us define the following two decision problems.

Definition 2 *The* CutFeasibility *problem is as follows.*

Input *A warehouse multiset* $W = \{w_1, w_2, \ldots, w_n\}$ *and an orders multiset* $N = \{(a_1, b_1), \ldots, (a_m, b_m)\}$.

Question *Is there a valid cutting assignment?*

In the other problem it is assumed that a valid cutting assignment exists.

Definition 3 *The* MaxExactFit *problem is as follows.*

Input *A warehouse multiset* $W = \{w_1, w_2, \ldots, w_n\}$ *and an orders multiset* $N = \{(a_1, b_1), \ldots, (a_m, b_m)\}$ *such that a valid cutting assignment for* W *and* N *exists, furthermore a natural number* k.

Question *Is there a valid cutting assignment with at least* k *exact fits?*

For both of these problems it is clear that they are in NP, since a valid cutting assignment or one with the desired property is a good witness.

Proposition 4 CutFeasibility *is NP-complete.*

In fact, BinPacking is a special case of CutFeasibility.

Theorem 5 MaxExactFit *is also NP-complete.*

Although BinPacking is closely related to MaxExactFit, their optima may be attained in different cases-

Proposition 6 *There are examples of set of weights* $s_1, s_2, \ldots, s_m$ *such that optimal solution for* BinPacking *uses less bins of capacity one, than the optimal solution for* MaxExactFit *in case of unit length rods in the warehouse and* $a_i = b_i = s_i$ *for all* $1 \leq i \leq m$.

In fact the difference in the Proposition above can be arbitrary large. We have to be careful how the problem is formulated. It is important that we want to minimize the number of cuts, that is maximize the number of exact fits in a valid cutting assignment.

Proposition 7 *Maximum number of exact fits are not necessarily given by a valid cutting assignment, even if such an assignment exists.*

3 Practical approaches

3.1 Dynamic programming and clique search

Definition 8 *Let P be a set of orders and w be an element from the warehouse. Define*

$$fit(P,w) = \begin{cases} 1 & if\ \sum_{(a_i,b_i)\in P} a_i \le w \le \sum_{(a_i,b_i)\in P} b_i \\ 0 & otherwise \end{cases}$$

Define a compatibility graph G for order sets and warehouse sets as the following

$$V(G) = \{(P,w) : fit(P,w) = 1\}$$
$$E(G) = \{\{(P_i,w_i),(P_j,w_j)\} \colon P_i \cap P_j = \varnothing \text{ and } w_i \neq w_j\}$$

1. Identify all subsets of orders that match a certain rod *exactly*, taking into account the tolerances using *dynamic programming*.
2. Construct the compatibility graph and find the largest compatible set of sets that is a *largest clique* in G.

This method is ineffective, the size of the compatibility graph becomes too large even for moderate sized inputs.

3.2 $0-1$-linear programming

Define indicator variables x_{ji} by

$$x_{ji} = \begin{cases} 1 & \pi(a_j,b_j) = w_i \\ 0 & \text{otherwise} \end{cases}$$

Then we have the following set of constraints:

(i) $\forall i \colon \sum_j a_j x_{ji} \le w_i$,

(ii) $\forall i \colon \sum_j b_j x_{ji} \ge \lambda_i w_i$,

(iii) $\forall j \colon \sum_i x_{ji} = 1$,

where $x_{ji}, \lambda_i \in \{0,1\}$. Condition (i) states that the orders assigned to rod i fit to that rod, (ii) means that if $\lambda_i = 1$ then we have an exact fit, while (iii) is to assure that the cutting assignment is valid, that is every order is assigned to exactly one rod. We want to maximize $\sum_i \lambda_i$. We used the Gurobi Solver [**?**] to solve test cases. We found that it is much more effective than the compatibility graph method. Some test run data is shown in Table 1.

3.3 Hierarchical optimization

Since usually the minimum cutting assignment is not unique, there is a possibility for another optimization. Our goal is to achieve longer pieces of leftovers from the rods cut as they are better usable, than short ones. In order to do so we set as secondary goal to maximize $\sum_i (w_i - \sum_j a_j x_{ji})^2 (1-\lambda_i)$, which is the sum of the squares of lengths of the leftover pieces, as $\lambda_i = 1$ iff there is an exact fit on rod i. However, only linear goal functions are allowed in hierarchical optimization by Gurobi, so this function must be "linearized". We have terms $x_{ji}^2, x_{ji}x_{ki}, x_{ji}\lambda_i, x_{ji}x_{ki}\lambda_i$ that are not linear. We use that our variables can only take values from $\{0,1\}$ so x_{ji}^2 is replaced by x_{ji}, and new variables $x_{jki} = x_{ji} \wedge x_{ki}$ and $L_{jki} = x_{ji} \wedge x_{ki} \wedge \lambda_i$ are introduced for the other products. These new variable definitions are added as constraints for the optimization, since Gurobi has the logical `AND` function built in.

Table 1: Test run data using Gurobi Solver

warehouse size	order size	time (s)
12	21	0.0369
14	25	0.0316
16	28	0.0903
18	32	0.1203
20	36	0.1326
24	43	0.2744
26	46	0.1269
28	50	0.2750
32	57	0.5966
34	61	0.7091
36	64	1.3148
38	68	1.5359
40	72	2.4816
42	75	1.0035
44	79	1.0154
46	82	4.7756

References

[1] U. Čibej, Personal communication,

[2] `https://www.gurobi.com/`

Partitioning into common independent sets via relaxing strongly base orderability

Kristóf Bérczi[1]

MTA-ELTE Matroid Optimization Research Group
ELKH-ELTE Egerváry Research Group
Department of Operations Research
Eötvös Loránd University
Budapest, Hungary
kristof.berczi@ttk.elte.hu

Tamás Schwarcz[1,2]

MTA-ELTE Matroid Optimization Research Group
Department of Operations Research
Eötvös Loránd University
Budapest, Hungary
tamas.schwarcz@ttk.elte.hu

Abstract: The problem of covering the ground set of two matroids by a minimum number of common independent sets is notoriously hard even in very restricted settings, i.e. when the goal is to decide if two common independent sets suffice or not. Nevertheless, as the problem generalizes several long-standing open questions, identifying tractable cases is of particular interest. Strongly base orderable matroids form a class for which a basis-exchange condition that is much stronger than the standard axiom is met. As a result, several problems that are open for arbitrary matroids can be solved for this class. In particular, Davies and McDiarmid showed that if both matroids are strongly base orderable, then the covering number of their intersection coincides with the maximum of their covering numbers.

Motivated by their result, we propose relaxations of strongly base orderability in two directions. First we weaken the basis-exchange condition, which leads to the definition of a new, complete class of matroids with distinguished algorithmic properties. Second, we introduce the notion of covering the circuits of a matroid by a graph, and consider the cases when the graph is ought to be 2-regular or a path. We give an extensive list of results explaining how the proposed relaxations compare to existing conjectures and theorems on coverings by common independent sets.

Keywords: Coverings, Excluded minors, Matroid intersection, Matroidally k-colorability, Strongly base orderable matroids

1 Introduction

For basic definitions and notation of matroid theory, the interested reader is referred to [21]. Throughout the paper, we denote the ground set of a matroid M by E with $|E| = n$, while the sets of independent sets, bases and circuits are denoted by $\mathcal{I}$, $\mathcal{B}$ and $\mathcal{C}$, respectively. The **covering number** $\beta(M)$ of a matroid M is the minimum number of independent sets needed to cover its ground set. A matroid is then called **k-coverable** if $\beta(M) \leq k$. Whenever investigating the covering number, we assume the matroid to be loopless as otherwise the ground set obviously cannot be covered by independent sets. The value of $\beta(M)$ can be determined using the rank formula of the union of matroids due to Edmonds and Fulkerson [10].

[1]The work was supported by the Lendület Programme of the Hungarian Academy of Sciences – grant number LP2021-1/2021 and by the Hungarian National Research, Development and Innovation Office – NKFIH, grant numbers FK128673 and TKP2020-NKA-06.

[2]Tamás Schwarcz was supported by the ÚNKP-22-3 New National Excellence Program of the Ministry for Culture and Innovation from the source of the National Research, Development and Innovation Fund.

It is quite natural to consider an analogous notion for the intersection of two matroids. Given two matroids M_1 and M_2 on the same ground set, the **covering number $\beta(M_1 \cap M_2)$ of their intersection** is the minimum number of common independent sets needed to cover the common ground set. Determining the exact value of $\beta(M_1 \cap M_2)$ has been the center of attention for a long time since it generalizes a wide list of fundamental questions from both graph and matroid theory, including Woodall's conjecture [26] on the maximum number of pairwise disjoint dijoins in directed graphs, or Rota's basis conjecture [15] on packing transversal bases. Nevertheless, apart from partial results such as Kőnig's 1-factorization theorem [18] or Edmonds' disjoint arborescences theorem [9], the problem remained open until recently, when the authors settled the complexity of the problem by showing hardness under the rank oracle model [4].

As determining the exact value of $\beta(M_1 \cap M_2)$ is hard in general, the need for good lower and upper bounds arises. A lower bound is easy to give as $\beta(M_1 \cap M_2) \geq \min\{\beta(M_1), \beta(M_2)\}$ always holds. Nevertheless, the equality $\beta(M_1 \cap M_2) = \max\{\beta(M_1), \beta(M_2)\}$ does not necessarily hold for general matroids, as shown by the well-known example where M_1 is the graphic matroid of a complete graph on four vertices and M_2 is the partition matroid[1] defined by the partition of its edges into three matchings, see [23] for details. As for the upper bound, Aharoni and Berger [2] showed by using techniques from topology that $\beta(M_1 \cap M_2) \leq 2\max\{\beta(M_1), \beta(M_2)\}$. Furthermore, they verified the slightly stronger statement that $\beta(M_1 \cap M_2) \leq \beta(M_1) + \beta(M_2)$ holds whenever one of $\beta(M_1)$ and $\beta(M_2)$ divides the other. Nevertheless, no example is known for which the true value would be close to the upper bound. In fact, Aharoni and Berger [3] conjectured the following, originally attributed to [2].

Conjecture 1 (Aharoni and Berger) *Let M_1 and M_2 be matroids on the same ground set.*

(1) If $\beta(M_1) \neq \beta(M_2)$, then $\beta(M_1 \cap M_2) = \max\{\beta(M_1), \beta(M_2)\}$.

(2) If $\beta(M_1) = \beta(M_2)$, then $\beta(M_1 \cap M_2) \leq \max\{\beta(M_1), \beta(M_2)\} + 1$.

The conjecture was verified only for $\beta(M_1) = \beta(M_2) = 2$ by Aharoni, Berger and Ziv [3]. In the same paper, the authors also showed that if $\beta(M_1) = 2$ and $\beta(M_2) = 3$, then $\beta(M_1 \cap M_2) \leq 4$ holds. For $\beta(M_1) = 2$ and $\beta(M_2) = k \geq 4$, the current best bound follows from the result of Aharoni and Berger [2] mentioned above: $\beta(M_1 \cap M_2) \leq k + 2$ if k is even, and $\beta(M_1 \cap M_2) \leq k + 3$ if k is odd.

Among the results related to Conjecture 1, the probably most important one is due to Davies and McDiarmid [7], who studied the class of strongly base orderable matroids. A matroid is called **strongly base orderable** if

(SBO) for any pair A, B of bases, there exists a bijection $\varphi\colon A \setminus B \to B \setminus A$ such that $(A \setminus X) \cup \varphi(X)$ is a basis for every $X \subseteq A \setminus B$.

It is worth mentioning that this implies $(B \cup X) \setminus \varphi(X)$ being a basis as well. Strongly base orderable matroids are interesting and important because we have a fairly good global understanding of their structure, while frustratingly little is known about the general case. In particular, Davies and McDiarmid showed that the covering number of the intersection of two strongly base orderable matroids coincides with the obvious lower bound.

Theorem 2 (Davies and McDiarmid) *Let M_1 and M_2 be strongly base orderable matroids on the same ground set. Then $\beta(M_1 \cap M_2) = \max\{\beta(M_1), \beta(M_2)\}$.*

As many matroid classes that naturally appear in combinatorial and graph optimization problems, e.g. gammoids, are strongly base orderable, Theorem 2 was a milestone result in the research on packing common independent sets. Recently, the theorem received a renewed interest when Abdi, Cornuéjols and Zlatin [1] successfully attacked special cases of Woodall's conjecture with its help. However, there are basic matroid classes that do not satisfy strongly base orderability, e.g. graphic or paving matroids.

[1] All partition matroids considered in the paper have all-ones upper bounds on the partition classes without explicitly mentioning it.

Our contribution. Our research was motivated by the following question: can strongly base orderability in Theorem 2 replaced with some weaker assumption so that the statement remains true? Or more generally, can we verify Conjecture 1 for some reasonably broad class of matroids? As partial answers to these problems, we propose two relaxations of strongly base orderability that weaken the original definition in different aspects.

First, in Section 2 we omit the condition for the bijection to go between $A \setminus B$ and $B \setminus A$, and allow repartitioning the multiset $A \cup B$ into two new bases A' and B' satisfying $A' \cap B' = A \cap B$ and $A' \cup B' = A \cup B$. We show that matroids satisfying the relaxed condition form a complete class. We also prove that Theorem 2 remains true when both matroids are from the proposed class, and thus we identify new scenarios when packing problems are tractable. Finally, we explain how the new class fits in the hierarchy of existing matroid classes.

Second, in Section 3 we relax the condition of finding a bijection between $A \setminus B$ and $B \setminus A$. Instead, we seek for a graph whose vertex set coincides with the symmetric difference $A \triangle B$ of the bases, and the graph represents the matroid locally in the sense that every stable set in it corresponds to an independent set of the original matroid. In particular, we prove that the the graph in question can be chosen to be a path for graphic and paving matroids. We conjecture that an analogous statement hold for arbitrary matroids, and discuss a series of corollaries that would follows from such a result.

Due to space constraints, some proofs, details and further results are deferred to the full version of this paper.

2 Relaxing the fixed bases: strongly base reorderable matroids

When considering the extendability of Theorem 2 to a broader class of matroids, some natural candidates are immediate. A matroid is called **base orderable** if

(BO) for any pair A, B of bases, there exists a bijection $\varphi\colon A \setminus B \to B \setminus A$ such that $A - x + \varphi(x)$ and $B + x - \varphi(x)$ are bases for every $x \in A \setminus B$.

Clearly, strongly base orderable matroids are also base orderable since the bijection appearing in (SBO) satisfies (BO) as well. It is known that $M(K_4)$, the graphic matroid of a complete graph on four vertices is a forbidden minor for base orderability. Hence every base orderable matroid is contained in the class $\mathrm{Ex}(M(K_4))$ of $M(K_4)$-minor-free matroids. In fact, these inclusions are known to be strict, hence $\mathrm{SBO} \subsetneq \mathrm{BO} \subsetneq \mathrm{Ex}(M(K_4))$ hold.

Davies and McDiarmid [7] posed the problem of whether Theorem 2 remains true if strongly base orderability is replaced with the weaker assumption that both matroids are base orderable. Though this specific question remains open, the following example shows that both matroids being in $\mathrm{Ex}(M(K_4))$ does not suffice. The examples uses J, a self-dual rank-4 matroid introduced by Oxley [22].

Remark 3 We show that there exists a partition matroid M for which $\beta(J) = \beta(M) = 2$ and $\beta(J \cap M) = 3$. Consider the geometric representation of the matroid J on Figure 1; for the definition, see also [21, page 650]. Let M be the partition matroid defined by partition classes $\{a, h\}, \{b, g\}, \{c, e\}, \{d, f\}$, implying $\beta(J) = \beta(M) = 2$.

To see that $\beta(J \cap M) > 2$, suppose to the contrary that $E = B_1 \cup B_2$ is a decomposition into two common bases of J and M. Without loss of generality, we may assume that $h \in B_1$. As B_1 is a common basis, it contains exactly one element from each of the pairs $\{b, g\}$, $\{c, e\}$, $\{d, f\}$ and at most one element from each of the pairs $\{b, c\}$, $\{d, g\}$, $\{e, f\}$, hence $B_1 = \{b, d, e, h\}$ or $B_1 = \{c, f, g, h\}$. In either case, B_2 is not a basis of J, a contradiction. Therefore $\beta(J \cap M) > 2$, and so $\beta(J \cap M) = 3$ by the result of Aharoni, Berger and Ziv mentioned earlier.

Motivated by the proof of Theorem 2, we call a matroid **strongly base reorderable** (or SBRO for short) if

(SBRO) for any pair A, B of bases, there exists bases A', B' and a bijection $\varphi\colon A' \setminus B' \to B' \setminus A'$ such that $A' \cap B' = A \cap B$, $A' \cup B' = A \cup B$, and $(A' \setminus X) \cup \varphi(X)$ is a basis for every $X \subseteq A' \setminus B'$.

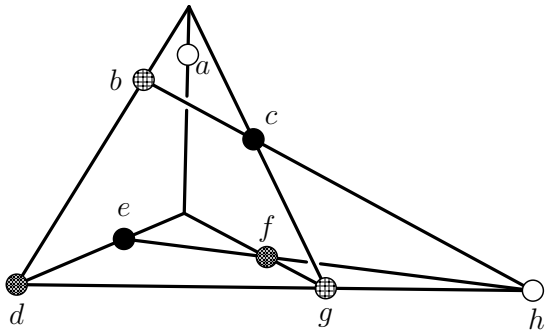


Figure 1: Geometric representation of the matroid J: bases are the sets of size four which do not lie on a plane. If $M_1 := J$ and M_2 is the partition matroid defined by partition classes $\{a,h\},\{b,g\},\{c,e\},\{d,f\}$ with upper bounds one, then $\beta(M_1) = \beta(M_2) = 2$ and $\beta(M_1 \cap M_2) = 3$.

In other words, (SBRO) differs from (SBO) in that it allows the repartitioning of the multiunion of the bases before asking for the bijection φ. It immediately follows from the definition that strongly base orderable matroids are strongly base reorderable as well.

Ingleton [17] defined a class of matroids to be **complete** if it is closed under taking minors, duals, direct sums, truncations and induction by directed graphs. A complete class is also closed under many other matroid operations, such as series and parallel connections, 2-sums, unions and principal extensions, see e.g. [5]. It was already noted by Ingleton [17] that the classes of base orderable and strongly base orderable matroids are complete, while Sims [24] verified that $\mathrm{Ex}(M(K_4))$ is complete as well. Bonin and Savitsky [5] showed that a class of matroids is complete if it is closed under minors, duals, direct sums and principal extensions, and they used this fact to verify that the class of so-called k-base-orderable matroids is complete for any fixed $k \geq 1$. The next theorem shows that SBRO matroids also form a complete class.

Theorem 4 SBRO *is a complete class.* □

With the help of Theorem 4, one can verify the next theorem using an analogous proof to that of Theorem 2 appearing in [23, Theorem 42.13].

Theorem 5 *Let M_1 and M_2 be strongly base reorderable matroids on the same ground set. Then $\beta(M_1 \cap M_2) = \max\{\beta(M_1), \beta(M_2)\}$.* □

In general, knowing the excluded minors for a minor-closed matroid class provides a powerful tool that then can be used in various applications. Based on the characterization of $M(K_4)$-minor-free binary matroids by Brylawski [6] and of $M(K_4)$-minor-free ternary matroids by Oxley [22], we give a characterization of binary and of ternary SBRO matroids.

Theorem 6 *The matroids $M(K_4)$ and J are excluded minors for* SBRO. *Furthermore,*

(a) a binary matroid is SBRO if and only if it does not contain $M(K_4)$ as a minor, and

(b) a ternary matroid is SBRO if and only if it does not contain $M(K_4)$ or J as a minor.

Proof: First we show that neither $M(K_4)$ nor J is SBRO. This follows from Theorem 5 as each $M_1 \in \{M(K_4), J\}$ is 2-coverable and there exists a 2-coverable partition matroid M_2 on the same ground set such that $\beta(M_1 \cap M_2) = 3$. Indeed, for $M_1 = M(K_4)$, let the three partition classes defining M_2 be the matchings of size two of K_4, see [23]. For $M_1 = J$, the statement follows by Remark 3.

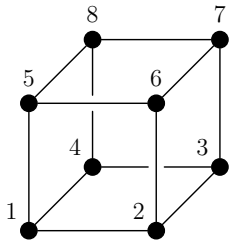


Figure 2: The rank-4 matroid $AG(3,2)$ in which each set of size three is independent, and the dependent sets of size four are the six *faces* of the cube, the six *diagonal planes* and the two *twisted planes* $\{1,3,6,8\}$ and $\{2,4,5,7\}$.

$M(K_4)$-minor-free binary matroids are SBRO, as they coincide with the graphic matroids of series-parallel graphs [6] which are strongly base orderable, see also [25]. As $M(K_4)$ is not SBRO, it follows that $M(K_4)$ is the unique binary excluded minor for SBRO.

Finally, we show that if M is a ternary excluded minor of SBRO distinct from $M(K_4)$, then M is isomorphic to J. As SBRO is closed under direct sums and 2-sums by Theorem 4, it follows that M is 3-connected. Oxley [22] showed that a 3-connected $M(K_4)$-minor-free ternary matroid is isomorphic either to the rank-r whirl $\mathcal{W}^r$ for some $r \geq 2$, to J, or to a minor of the matroid $S(5,6,12)$. Since $\mathcal{W}^r$ is a transversal matroid, it is strongly base orderable. The proof that $S(5,6,12)$ is SBRO is deferred to the full version of the paper. Therefore, M is isomorphic to J. As J is not SBRO and does not contain $M(K_4)$ as a minor, it follows that J is the unique ternary excluded minor for SBRO apart from $M(K_4)$. □

As $M(K_4)$ is the only binary, and $M(K_4)$ and J are the only ternary excluded minors for base orderability, Theorem 6 immediately implies the following.

Corollary 7 *A binary or ternary matroid is SBRO if and only if it is base orderable.*

Let us return to the open problem of Davies and McDiarmid on replacing strongly base orderability with base orderability in Theorem 2. Theorem 5 and Corollary 7 together imply an affirmative answer in the special case when the matroids are ternary – for binary matroids, this was already known as the classes of base orderable binary matroids and strongly base orderable binary matroids coincide. Motivated by this observation, it is natural to ask whether all base orderable matroids are SBRO. Unfortunately, the next example shows that this is not the case.

Remark 8 We construct a rank-5 matroid X_{10} on 10 elements that is base orderable but not SBRO. Recall the construction of the the binary affine cube $AG(3,2)$ from [21, page 645] using faces, diagonal planes and twisted planes, see Figure 2. We define X_{10} on the ground set $\{1,2,\ldots,8\} \cup \{a,b\}$ such that each set of size four is independent, and the family of dependent sets of size five is

$$\mathcal{H} \coloneqq \{F \cup \{a\} \mid F \text{ is a face}\} \cup \{F \cup \{b\} \mid F \text{ is a diagonal plane or a twisted plane}\}.$$

By the construction of paving matroids, see e.g. [14, Theorem 5.3.5], X_{10} is a paving matroid of rank 5 as $|H_1 \cap H_2| \leq 3$ for each $H_1, H_2 \in \mathcal{H}$, $H_1 \neq H_2$. It is not difficult to check that X_{10} does not contain $M(K_4)$ as a minor, hence it is base orderable since $M(K_4)$-minor-free paving matroids are base orderable [5]. The proof that X_{10} is not SBRO is deferred to the full version of the paper.

3 Relaxing the bijection: covering the circuits by a 2-factor

In this section, we propose another relaxation of strongly base orderability. In order to do so, first we give a new interpretation of property (SBO). Consider a matroid $M = (E, \mathcal{C})$ on ground set E with set

of circuits $\mathcal{C}$. Furthermore, assume that $G = (E, F)$ is a graph with vertex set E and edge set F. For a subset $X \subseteq E$, let $\mathcal{C}[X]$ denote the set of circuits of M that lie in X, that is, $\mathcal{C}[X] \coloneqq \{C \in \mathcal{C} \mid C \subseteq X\}$, and let $F[X]$ denote the set of edges of G induced by X. We say that G **covers** a subset $\mathcal{C}' \subseteq \mathcal{C}$ of circuits if $F[C] \neq \emptyset$ for every $C \in \mathcal{C}'$. In other words, every stable subset of E in G is such that it contains no circuit from $\mathcal{C}'$.

Using this terminology, (SBO) is equivalent to saying that for any pair A, B of bases of a strongly base orderable matroid $M = (E, \mathcal{C})$, there exists a graph G consisting of a matching between the elements of $A \setminus B$ and $B \setminus A$ that covers $\mathcal{C}[A \cup B]$. Similarly, property (SBRO) discussed in Section 2 translates into the existence of a matching on $A \triangle B$ that covers $\mathcal{C}[A \cup B]$. Observe the small but crucial difference between the two definitions: while the former asks for matching edges going between the elements of A and B, the latter allows the end vertices of any matching edge to fall in the same set, i.e. A or B.

We conjecture that a similar statement holds for arbitrary matroids where G is a 2-regular graph or a path instead of a matching, where a graph is **2-regular** if each vertex has degree exactly two. More precisely, we propose four relaxations of different strengths, and say that a matroid $M = (E, \mathcal{C})$ has property

(R) if for any pair A, B of bases, there exists a 2-regular graph on $A \triangle B$ that covers $\mathcal{C}[A \cup B]$,

(R+) if for any pair A, B of bases, there exists a 2-regular graph that consists of cycles alternating between $A \setminus B$ and $B \setminus A$ and covers $\mathcal{C}[A \cup B]$,

(P) if for any pair A, B of bases, there exists a path on $A \triangle B$ that covers $\mathcal{C}[A \cup B]$,

(P+) if for any pair A, B of bases, there exists a path that alternates between $A \setminus B$ and $B \setminus A$ and covers $\mathcal{C}[A \cup B]$.

In all cases, the condition that the graph in question covers $\mathcal{C}[A \cup B]$ is equivalent to requiring that the union of $A \cap B$ and any stable set in the graph is independent in M.

Since any path can be extended to a single cycle by adding an edge between its end vertices, property (P+) implies all the others, while property (R) is a special case of any of them. As for properties (R+) and (P), we could not show any connection between them. As a matching between $A \setminus B$ and $B \setminus A$ can always be extended to an alternating path between them, strongly base orderable matroids satisfy (P+), and an analogous reasoning shows that SBRO matroids satisfy (P).

3.1 Covering fundamental circuits

Observe that in all of properties (R) – (P+), the number of edges used to cover the circuits in question is bounded by $|A \triangle B|$. At this point, it is not even clear why those circuits could be covered by a small number of graph edges. As a first step towards understanding the general case, instead of covering every circuit, we concentrate on covering fundamental circuits only. Given a basis B of a matroid and an element a outside of B, we denote by $C_B(a)$ the **fundamental circuit** of a with respect to B, that is, the unique circuit in $B + a$. We extend this notation to sets as well, that is, for a set X disjoint from B we use $\mathcal{C}_B(X) \coloneqq \{C_B(x) \mid x \in X\}$.

Theorem 9 *Let A, B be bases of a matroid M.*

(a) There exists a 2-regular graph that consists of cycles alternating between $A \setminus B$ and $B \setminus A$ and covers $\mathcal{C}_A(B) \cup \mathcal{C}_B(A)$.

(b) There exists a tree that consists of edges between $A \setminus B$ and $B \setminus A$ and covers $\mathcal{C}_A(B) \cup \mathcal{C}_B(A)$.

Proof: The basis exchange axiom implies that there exists a bijection $\varphi_A \colon A \setminus B \to B \setminus A$ such that $A - a + \varphi_A(a)$ is a basis for each $a \in A \setminus B$, or equivalently, $a \in C_A(\varphi_A(a))$ (see e.g. [14, Theorem 5.3.4]). Similarly, there exists a bijection $\varphi_B \colon B \setminus A \to A \setminus B$ such that $B - b + \varphi_B(b)$ is a basis for each $b \in B \setminus A$, or equivalently, $b \in C_B(\varphi_B(b))$. Therefore the graph consisting of edges $\{a\varphi_A(a) \mid a \in A \setminus B\} \cup \{b\varphi_B(b) \mid b \in B \setminus A\}$ is a 2-regular graph that covers $\mathcal{C}_A(B) \cup \mathcal{C}_B(A)$, proving (a).

By the symmetric exchange axiom, there exists a mapping – not necessarily a bijection – $\phi_A \colon A \setminus B \to B \setminus A$ such that both $A - a + \phi_A(a)$ and $B + a - \phi_A(a)$ are bases for each $a \in A \setminus B$, or equivalently, the edge $a\phi_A(a)$ covers both $C_A(\phi_A(a))$ and $C_B(a)$. Similarly, there exists a mapping $\phi_B \colon B \setminus A \to A \setminus B$ such that both $B - b + \phi_B(b)$ and $A + b - \phi_B(b)$ are bases for each $b \in B \setminus A$, or equivalently, the edge $b\phi_B(b)$ covers both $C_B(\phi_B(b))$ and $C_A(b)$. Consider the graph consisting of edges $\{a\phi_A(a) \mid a \in A \setminus B\} \cup \{b\phi_B(b) \mid b \in B \setminus A\}$. Note that the graph may not be connected, but each vertex in $A \triangle B$ has degree at least one in it. Hence any maximum forest in this graph covers $\mathcal{C}_A(B) \cup \mathcal{C}_B(A)$. As extending the forest to a tree by adding edges connecting the components maintains this property, (b) follows. □

Theorem 9 shows that the fundamental circuits corresponding to the basis pair can be covered by a 2-regular graph. However, the same question remains open when the graph is required to be a path, and this seemingly simple task already appears to be highly non-trivial. The essence of part (b) of the theorem is that, instead of a path, the covering can always be realized by a tree.

3.2 Graphic and paving matroids

We could not identify any matroid for which (P+) would fail, hence we propose the following, probably overly optimistic conjecture.

Conjecture 10 *Let A, B be bases of a matroid M. Then there exists a path that alternates between $A \setminus B$ and $B \setminus A$ and covers $\mathcal{C}[A \cup B]$.*

To validate the conjecture somewhat, we show that the statement holds for graphic and paving matroids. We start with a technical lemma.

Lemma 11 *Let $\mathcal{M}$ be a minor-closed class of matroids and $(X) \in \{(R), (R+), (P), (P+)\}$. To verify that (X) holds for each $M \in \mathcal{M}$, it suffices to show that the property holds when the ground set is the disjoint union of A and B.*

Proof: Let A and B be bases of the matroid $M = (E, \mathcal{C})$ where $M \in \mathcal{M}$. If $A \cap B \neq \emptyset$ or $A \cup B \neq E$, then define $A' \coloneqq A \setminus B$, $B' \coloneqq B \setminus A$, $E' \coloneqq A' \cup B'$, and let $M' = (E', \mathcal{C}')$ denote the matroid obtained from M by deleting $E \setminus (A \cup B)$ and contracting $A \cap B$. Note that A' and B' are disjoint bases of M' whose union is E'. Furthermore, for any circuit $C \in \mathcal{C}[A \cup B]$, there exists a circuit $C' \in \mathcal{C}'[A' \cup B']$ such that $C' \subseteq C$. Therefore, any graph proving (X) for the pair A', B' in M' also proves (X) for the pair A, B in M. □

With the help of Lemma 11, we first verify the graphic case. It may be confusing that (P+) states the existence of a certain path and we are working with graphs, so let us emphasize that the path is defined on the elements of the ground set, i.e. the edges of the underlying graph, and it does not appear in the graph itself in any sense.

Theorem 12 *Graphic matroids have property (P+).*

Proof: Let $G = (V, E)$ be a graph with graphic matroid $M(G)$. We prove the theorem by induction on $|E|$. Note that (P+) clearly holds when $E = \emptyset$. Take a pair of bases A, B. As the class of graphic matroids is closed under taking minors, Lemma 11 applies, hence we may assume that E is the disjoint union of A and B. Furthermore, we may assume that G is connected as otherwise we can pick a vertex in each connected component and identify those, thus obtaining a connected graph with the same graphic matroid as the original one.

By the above, we have $|E| = |A| + |B| = 2(|V| - 1)$, hence G has a vertex v of degree at most three. If v has degree two, then G has an edge $a \in A$ and an edge $b \in B$ incident to v. Since $G' \coloneqq G - v$ is the union of disjoint spanning trees $A - a$ and $B - b$, there exists a path P' covering the circuits of $M(G')$ by the induction hypothesis. As every cycle of G passing through v uses the edges a and b, adding an

extra edge to $P' \cup \{ab\}$ between a and the endpoint of P' in B results in an alternating path between A and B covering the circuits of $M(G)$.

If v has degree three, then we may assume that edges $a_1, a_2 \in A$ and $b \in B$ are incident to v. Consider the graph G' obtained by contracting a_2 and deleting b, let A' denote the spanning tree of G' obtained from A by contracting a_2, and define $B' := B - b$. By the induction hypothesis, there exists a path P' alternating between A' and B' that covers the circuits of $M(G')$. Let c be a neighbour of a_1 in P' and consider the path $P := (P' - a_1c) \cup \{a_1b, ba_2, a_2c\}$. We claim that P covers every circuit C of $M(G)$. If C corresponds to a cycle of G that does not pass through v, then it is also a circuit of $M(G')$, and therefore it is covered by $P' - a_1c$. If $b \in C$, then either $a_1 \in C$ or $a_2 \in C$, hence C is covered either by a_1b or ba_2. The only remaining case is when $a_1, a_2 \in C$. Let C' denote the cycle of G' obtained by contracting a_2 in C. Then C' is covered by an edge of P'. If $c \notin C$, then this is an edge of $P' - a_1c$ as well, otherwise C is covered by the edge a_2c. This proves that P covers all circuits of $M(G)$. □

Remark 13 The fact that the path in (P+) is defined on the edges of the graph might be confusing, hence let us rephrase Theorem 12 using graph terminology as it might be of independent combinatorial interest. The theorem is equivalent to the following: For any graph that is the union of two spanning trees A and B, its edges can be ordered in such a way that the elements of A and B appear alternately, and every cycle of G contains two consecutive elements.

A rank-r matroid is called **paving** if each set of size at most $r-1$ is independent, or in other words, each circuit has size r or $r+1$.

Theorem 14 *Paving matroids have property (P+).*

Proof: We prove the theorem by induction on $|E|$. Note that (P+) clearly holds when $E = \emptyset$. Take a pair of bases A, B. As the class of graphic matroids is closed under taking minors, Lemma 11 applies, hence we may assume that E is the disjoint union of A and B.

Let $a_r \in A$ and $b_r \in B$ such that $A - a_r + b_r$ is a basis. Then $A - a_r$ and $B - b_r$ are bases of the paving matroid M' obtained from M by contracting b_r and deleting a_r. By the induction hypothesis, there exist orderings $a_1, \ldots, a_{r-1}$ of $A - a_r$ and $b_1, \ldots, b_{r-1}$ of $B - b_r$ such that the path $P' = a_1, b_1, \ldots, a_{r-1}, b_{r-1}$ covers the circuits of M'. We claim that the path $P = a_1, b_1, \ldots, a_r, b_r$ covers each circuit of M. Since M is paving, we only need to consider circuits C of size r as circuits of size $r+1$ are clearly covered. If $a_r \notin C$, then $C - b_r$ contains a circuit of M' which is covered by P'. If $a_r \in C$, then C is covered by P as A is the only stable set of size r containing a_r. Therefore, P covers every circuit of M. □

3.3 Applications for covering problems

Motivated by a conjecture of Du, Hsu and Hwang [8], Erdős [11] popularized the so-called *cycle plus triangles problem* which asked whether every 4-regular graph that is the edge-disjoint union of a Hamiltonian cycle and pairwise vertex-disjoint triangles is 3-colorable. Fleischner and Stiebitz [12] answered this question in the affirmative. In the past decades, their work was followed by a series of papers that studied related problems, summarized below.

Theorem 15 (Fleischner and Stiebitz, McDonald and Puleo, Haxell)

(a) If a graph G is the edge-disjoint union of a Hamiltonian cycle and some pairwise vertex-disjoint triangles, then G is 3-colorable.

(b) If $k \geq 4$ and a graph G is the edge-disjoint union of a Hamiltonian cycle and some pairwise vertex-disjoint complete subgraphs each on at most k vertices, then G is k-colorable.

(c) If $k \geq 4$ and a graph G is the edge-disjoint union of a 2-regular bipartite graph and some pairwise disjoint cliques each on at most k vertices, then G is k-colorable.

(d) If $k \geq 5$ and a graph G is the edge-disjoint union of a 2-regular graph and some pairwise vertex-disjoint complete subgraphs each on at most k vertices, then G is k-colorable.

Statement (b) was proven by Fleischner and Stiebitz [13]. McDonald and Puleo [20] verified for $k \geq 4$ that if a graph is decomposable into cliques on exactly k vertices and a 2-regular graph with at most one odd cycle of length exceeding three, then it is k-colorable. This implies statement (c), as we can extend the cliques having less than k vertices and the 2-regular bipartite graph by adding extra vertices and edges to ensure that each clique has exactly k vertices. Finally, statement (d) follows from the result of Haxell [16] that a graph is k-colorable if it is decomposable into cliques on k vertices and a graph H such that $k \geq 3\Delta(H) - 1$. Note that the complete graph on four vertices shows that statements (b) and (c) do not hold for $k = 3$, while it is open whether (d) holds for $k = 4$.

Theorem 16 *Let $M_1 = (E, \mathcal{I}_1)$ be a 2-coverable matroid and $M_2 = (E, \mathcal{I}_2)$ be a k-coverable partition matroid.*

(a) If $k \geq 3$ and M_1 satisfies (P), then $\beta(M_1 \cap M_2) \leq k$.

(b) If $k \geq 4$ and M_1 satisfies (R+), then $\beta(M_1 \cap M_2) \leq k$.

(c) If $k \geq 5$ and M_1 satisfies (R), then $\beta(M_1 \cap M_2) \leq k$.

Proof: To prove (a), let $E = A \cup B$ be a decomposition of M_1 into two bases and P be a path on $A \triangle B$ covering the circuits of M_1. Let $E_1, \ldots, E_q$ denote the partition classes of M_2 and let Q denote the graph obtained by taking the union of complete graphs on E_i for $i = 1, \ldots, q$. We claim that the graph $G := P \cup Q$ is k-colorable. Indeed, if $k = 3$, then G is a subgraph of a graph appearing in Theorem 15(a), while for $k \geq 4$ we can apply Theorem 15(b) after adding an extra edge to G between the end vertices of P. As each stable set of P is independent in M_1 and each stable set of Q is independent in M_2, the k-colorability of G implies that $\beta(M_1 \cap M_2) \leq k$.

Parts (b) and (c) can be proven analogously by applying Theorem 15(c) and (d), respectively. □

One of the main consequences of the results discussed so far is that we confirm Conjecture 1 for instances that were not settled before.

Corollary 17 *Conjecture 1 holds if M_1 is either a graphic matroid or a paving matroid with $\beta(M_1) = 2$, and M_2 is a partition matroid.*

Proof: The conjecture was settled for $k \leq 2$ in [3]. For $k \geq 3$, the statement follows by combining Theorems 12–14 and Theorem 16(a). □

Kotlar and Ziv [19] defined an element e of a matroid M to be $(k+1)$**-spanned** if there exist $k+1$ pairwise disjoint sets such that e is spanned by each of them in M. This condition is equivalent to the existence of k pairwise disjoint sets not containing e but spanning it, as one of the sets can be chosen to be $\{e\}$. If a matroid does not contain any $(k+1)$-spanned element, then it is not difficult to show that it is k-coverable. Kotlar and Ziv conjectured that if no element is $(k+1)$-spanned in either M_1 or M_2, then $\beta(M_1 \cap M_2) \leq k$. They verified the conjecture if $k = 2$ or the ground set decomposes into k bases in each of M_1 and M_2. Next we show how the absence of $(k+1)$-spanned elements can be combined with (R) or (P).

Theorem 18 *Let $M_1 = (E, \mathcal{I}_1)$ be a 2-coverable matroid and $M_2 = (E, \mathcal{I}_2)$ be a matroid with no $(k+1)$-spanned elements.*

(a) If M_1 satisfies (P), then $\beta(M_1 \cap M_2) \leq k + 1$.

(b) If M_1 satisfies (R), then $\beta(M_1 \cap M_2) \leq k + 2$.

PROOF: To prove (a), let $E = \{e_1, \ldots, e_n\}$ be such that every stable set of the path defined by the ordering $e_1, \ldots, e_n$ is independent in M_1. Color the elements in the order $e_1, \ldots, e_n$ greedily with positive integers such that an element e_i receives the smallest color c which is distinct from the color of e_{i-1}, and the set of elements already having color c does not span e_i in M_2. This procedure results in a coloring such that each color class is independent in M_2 and form a stable set of the path, thus it is independent in M_1 as well. As M_2 contains no $(k+1)$-spanned elements, the number of colors used is at most $k+1$. This proves that $\beta(M_1 \cap M_2) \leq k+1$.

Statement (b) follows by a similar greedy argument. □

Aharoni and Berger [2] defined a graph G to be **matroidally k-colorable** if for every k-coverable matroid M on the vertex set of G, the ground set can be decomposed into k stable sets of G which are independent in M. The following facts are not difficult to show for matroidally k-colorable graphs.

Lemma 19

(a) A subgraph of a matroidally k-colorable graph is matroidally k-colorable.

(b) A matroidally k-colorable graph is matroidally $(k+1)$-colorable.

PROOF: The proof of (a) is straightforward. To prove (b), let G be a matroidally k-colorable graph on vertex set V and M be a $(k+1)$-coverable matroid on V. Let $V = I_1 \cup \cdots \cup I_{k+1}$ be a partition of V into independent sets of M. As the graph $G[I_1 \cup \ldots I_k]$ is matroidally k-coverable by (a) and $M \setminus I_{k+1}$ is a k-coverable matroid on its vertex set, there exist stable sets $S_1, \ldots, S_k$ of G which are independent in $M \setminus I_{k+1}$ such that $S_1 \cup \cdots \cup S_k = I_1 \cup \cdots \cup I_k$. Since $G[S_2 \cup \cdots \cup S_k \cup I_{k+1}]$ is matroidally k-coverable by (a) and $M \setminus S_1$ is a k-coverable matroid on its vertex set, there exist stable sets $S'_2, \ldots, S'_{k+1}$ of G which are independent in $M \setminus S_1$ such that $S'_2 \cup \cdots \cup S'_{k+1} = S_2 \cup \cdots \cup S_k \cup I_{k+1}$. Then $V = S_1 \cup S'_2 \cup \cdots \cup S'_{k+1}$ is a partition of V into stable sets of G which are independent in M. □

As a generalization of Theorem 15(a), Aharoni and Berger [2] conjectured that the cycle $C_{3\ell}$ is matroidally 3-colorable for every $\ell \geq 1$. Their conjecture, when combined with Lemma 19(a), would imply the following.

Conjecture 20 *Every path is matroidally 3-colorable.*

Remark 21 It is reasonable to ask whether there is a connection between Conjectures 1 and 20. It turns out that the latter implies the former when M_1 is 2-coverable matroid satisfying (P).

Indeed, let P be a path on vertex set E such that every stable set of P is independent in M_1, and let $M_2 = (E, \mathcal{I}_2)$ be an arbitrary k-coverable matroid. If $k = 2$, then $\beta(M_1 \cap M_2) \leq 3$ holds by the results of Aharoni, Berger and Ziv [3]. If $k \geq 3$ and Conjecture 20 holds, then P is matroidally k-colorable by Lemma 19(b), hence E can be decomposed into k stable sets of P which are independent set in M_2. This gives a decomposition of E into k common independent sets of M_1 and M_2.

By the above reasoning, Conjectures 10 and 20 together would imply Conjecture 1 when M_1 is 2-coverable.

Finally, let us mention an interesting result on the intersection of q matroids satisfying (R). Let $M_1, \ldots, M_q$ be 2-coverable matroids over the same ground set. In general, not much is known about the minimum number of common independent sets $\beta(M_1 \cap \ldots, \cap M_q)$ needed to cover their ground set. An obvious upper bound is the product of their individual covering numbers, that is, $\beta(M_1 \cap \cdots \cap M_q) \leq \prod_{i=1}^{q} \beta(M_i) = 2^q$. Nevertheless, a much stronger upper bound follows if each M_i satisfies (R).

Theorem 22 *Let $M_1, \ldots, M_q$ be 2-coverable matroids satisfying (R). Then $\beta(M_1 \cap \cdots \cap M_q) \leq 2q+1$.*

PROOF: Let G_i be a 2-regular graph covering the circuits of M_i for $1 \leq i \leq q$. Since the graph $G \coloneqq G_1 \cup \cdots \cup G_q$ has maximum degree at most $2q$, it is $(2q+1)$-colorable. Since every stable set of G is stable in each G_i, this coloring gives a decomposition of the ground set into $2q+1$ common independent sets of the matroids. □

References

[1] A. Abdi, G. Cornuéjols, and M. Zlatin. On packing dijoins in digraphs and weighted digraphs. *arXiv preprint arXiv:2202.00392*, 2022.

[2] R. Aharoni and E. Berger. The intersection of a matroid and a simplicial complex. *Transactions of the American Mathematical Society*, 358(11):4895–4917, 2006.

[3] R. Aharoni, E. Berger, and R. Ziv. The edge covering number of the intersection of two matroids. *Discrete Mathematics*, 312(1):81–85, 2012.

[4] K. Bérczi and T. Schwarcz. Complexity of packing common bases in matroids. *Mathematical Programming*, 188(1):1–18, 2021.

[5] J. E. Bonin and T. J. Savitsky. An infinite family of excluded minors for strong base-orderability. *Linear Algebra and its Applications*, 488:396–429, 2016.

[6] T. H. Brylawski. A combinatorial model for series-parallel networks. *Transactions of the American Mathematical Society*, 154:1–22, 1971.

[7] J. Davies and C. McDiarmid. Disjoint common transversals and exchange structures. *Journal of the London Mathematical Society*, 2(1):55–62, 1976.

[8] D. Du, D. Hsu, and F. Hwang. The Hamiltonian property of consecutive-d digraphs. *Mathematical and Computer Modelling*, 17(11):61–63, 1993.

[9] J. Edmonds. Edge-disjoint branchings. In *Combinatorial Algorithms*. Academic Press, New York, 1973.

[10] J. Edmonds and D. R. Fulkerson. Transversals and matroid partition. *Journal of Research of the National Bureau of Standards (B)*, 69:147–153, 1965.

[11] P. Erdős. On some of my favourite problems in graph theory and block designs. *Le Matematiche*, 45(1):61–74, 1990.

[12] H. Fleischner and M. Stiebitz. A solution to a colouring problem of P. Erdős. *Discrete Mathematics*, 101(1-3):39–48, 1992.

[13] H. Fleischner and M. Stiebitz. Some remarks on the cycle plus triangles problem. In *The Mathematics of Paul Erdös II*, pages 136–142. Springer, Berlin, 1997.

[14] A. Frank. *Connections in Combinatorial Optimization*, volume 38 of *Oxford Lecture Series in Mathematics and its Applications*. Oxford University Press, Oxford, 2011.

[15] J. Geelen and K. Webb. On Rota's basis conjecture. *SIAM Journal on Discrete Mathematics*, 21(3):802–804, 2007.

[16] P. E. Haxell. On the strong chromatic number. *Combinatorics, Probability and Computing*, 13(6):857–865, 2004.

[17] A. W. Ingleton. Transversal matroids and related structures. In *Higher Combinatorics*, pages 117–131. Springer Netherlands, 1977.

[18] D. König. Über Graphen und ihre Anwendung auf Determinantentheorie und Mengenlehre. *Mathematische Annalen*, 77(4):453–465, 1916.

[19] D. Kotlar and R. Ziv. On partitioning two matroids into common independent subsets. *Discrete Mathematics*, 300(1-3):239–244, 2005.

[20] J. McDonald and G. J. Puleo. Strong coloring 2-regular graphs: Cycle restrictions and partial colorings. *Journal of Graph Theory*, 100(4):653–670, 2022.

[21] J. Oxley. *Matroid Theory*, volume 21 of *Oxford Graduate Texts in Mathematics*. Oxford University Press, Oxford, second edition, 2011.

[22] J. G. Oxley. A characterization of the ternary matroids with no $M(K_4)$-minor. *Journal of Combinatorial Theory, Series B*, 42(2):212–249, 1987.

[23] A. Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*. Springer, Berlin, 2003.

[24] J. A. Sims. A complete class of matroids. *The Quartefrly Journal of Mathematics*, 28(4):449–451, 1977.

[25] D. J. A. Welsh. *Matroid Theory*. Academic Press, London, 1976.

[26] D. R. Woodall. Menger and Kőnig systems. In *Theory and Applications of Graphs*, pages 620–635. Springer, Berlin, 1978.

Jump-systems of T-paths

Mouna Sadli

Institute for Higher Education in Morocco
Avenue Mohamed VI, Km 4.2
(Route des Zaërs) Souissi, Rabat, Morocco
msadli@iihem.ac.ma

András Sebő

Combinatorial Optimization
Univ. Grenoble-Alpes, CNRS, G-SCOP
46 Avenue Félix Viallet, 38240 Grenoble, France
Andras.Sebo@cnrs.fr

Abstract: Jump systems are sets of integer vectors satisfying a simple axiom, generalizing matroids, also delta-matroids, and well-kown combinatorial examples such as degree sequences of subgraphs of a graph. It is useful to know if a set of vectors defined from combinatorial structures is a jump system: this has consequences for optimizing on the set, or on some derived sets of vectors. In this note we are mainly concerned in telling our proof of the following more than two decades old fact and its original, elementary proof for an example different from degree sequences:

Given an udirected graph $G = (V, E)$ and $T \subseteq V$, the vectors m indexed by T for which there exist a set of openly disjoint T-paths so that each $t \in T$ is the endpoint of exactly $m(t)$ paths forms a jump system. The same holds for edge-disjoint T-paths.

We are also exhibiting the context and some consequences of this fact, with some pointers to recent developments, among them another proof by Iwata and Yokoi in this volume, and to some open problems.

Keywords: routing, disjoint paths, Mader's theorem, jump systems, bisubmodular polyhedra

1 Introduction

For basic notations and terminology we refer to Schrijver [20]. Given an undirected graph, $G = (V, E)$, and $T \subseteq V$, a T-path is a path P such that $V(P) \cap T$ consists of the two endpoints of P. Two T-paths, P, Q are *openly disjoint*, if $V(P) \cap V(Q) \subseteq T$, and they are edge-disjoint if $E(P) \cap E(Q) = \emptyset$. An integer vector $m \in \mathbb{Z}^n$ is called *vertex-feasible* or *edge-feasible* (for (G, T)) if there exists a set of openly vertex-, resp. edge-disjoint T-paths so that each $t \in T$ is the endpoint of $m(t)$ of them. Edge-feasible vectors are also called *node-demands*, and have been studied in [7], based on which the convex hull of edge-feasible vectors can be determined. We define jump systems on T:

A set $J \subseteq \mathbb{Z}^T$ is a *jump system*, if for each pair $x, y \in J$ and any step x' from x to y, either $x' \in J$, or there exists a step x'' from x' to y so that $x'' \in J$. A *step* x' from x to y is $x' := x$ if $x = y$, or $x' := x + e_s$ for some $s \in T$ such that $x_s < y_s$, or $x' := x - e_s$ for some $s \in \{1, \ldots, n\}$ such that $x_s > y_s$. The *unit vector* $e_s \in \{0, 1\}^T$ is defined by $e_s(s) = 1$ and $e_s(t) = 0$ if $t \in T \setminus \{s\}$.

This simple notion has been defined by Bouchet and Cunningham [2], generalizing the delta-matroid-axioms defined earlier by Bouchet [1], themselves generalizing matroid axioms. Besides (delta-)matroids – the $0-1$ special case –, one of the best-known examples of jump-systems presented in [2] are the degree sequences of graphs. Another example occurred to us in 1999-2000 [19]: feasible vectors for sets of openly vertex- or edge-disjoint T- paths.

Then under the impact of Schrijver's simple proof [21] of Mader's theorem [16] and following his proof of the matroid property of (inclusionwise) maximal feasible vectors in a slightly different, but equivalent, $0-1$ context – presented at the winter-school "New Methods in Discrete Mathematics" in Alpes d'Huez, March 2000 –, we have proved that sets of vertex- and edge-feasible vectors form actually jump systems.

The publication of our proof now was encouraged by some renewed interest and deep results concerning jump systems and their intersections. First, related to T-paths, by Iwata and Yokoi [9], and actually a draft of their proof of the jump-system property in it, in February 2022, that follows Lovász's method for proving Mader's theorem [16] which is thus very different from our proof following Schrijver proof; second, by Dudycz and Paluch's results [6] concerning weighted general graph factors, generalized by [11] to optimize on some particular weighted jump system intersections. These make worth summarizing some new and old connections in Section 3, with pointers to graph factors to analogous results for T-paths, and to common generalizations. Since our proof of the jump system property of T-paths is not easy to access (the only public access were lectures [19] and then a French thesis [18]) we decided to make it available in this note: it is in Section 2. Some of the not completely recent corollaries and the conjecture of Section 3 also keep the actuality of the subject with enhanced connections, until today.

2 The jump system of feasible vectors

Denote by $J_{\text{vertex}}(G,T)$ and $J_{\text{edge}}(G,T)$ the set of all vertex-feasible and edge-feasible vectors respectively.

Schrijver [20, Theorem 73.5, page 1292] considered the matroid property of (inclusionwise) maximal feasible vectors. The authors were lucky enough to hear this result and Schrijver's simple proof [21], [20, Theorem 73.2] of Mader's theorems [15], [16] – reducing them to a result of Gallai [8], itself shortly proved from Tutte's theorem on maximum matchings – and its corollaries, ahead of time, at the winter-school "New Methods in Discrete Mathematics" in Alpes d'Huez, March 2000. They were strongly interested, since in 1999 they have proved weaker results [19] in the same direction, first about the convex hull of $J_{\text{edge}}(G,T)$. We will discuss some still useful connections of these results in Section 3, with a related open problem.

Mader's theorem on the maximum number of edge-disjoint T-paths [15] is a straightforward consequence of the vertex-version [16] by taking the line graph, but no easy reduction is known in the other direction. For proving the matroid property or the jump-system property there is no need of neither theorems though, and there is no essential difference between the proof of the vertex- or edge-version. The proof of these jump-system properties is much simpler than that of Mader's theorems, and each of the vertex- or edge-versions can be obtained by mimicking the other.

Schrijver chooses the vertex-version for the proof of his matroid-property. We choose the edge version for a difference, for the sake of introducing the possibly useful idea of "edge-transitions" for edge-disjoint paths, and also because the theory of edge-disjoint paths has been much further developped than that of vertex-disjoint paths [10], [17], [4], [13], [12], [7]: capacities can then be put on edges, and T-paths can be generalized, the linear constraints for the "node demand polyhedron" of edge-disjoint paths have been determined, so there is more to say about the edge-version. (Similar weighted generalizations involving node-capacities are in principle possible though for the vertex-versions as well, but the corresponding decision problems are $\mathcal{NP}$-hard, and no natural analogue of the parity condition is known to ensure tractability.)

An *edge-transition* is an (unordered) pair of incident edges (equivalently, an edge of the line graph), or a pair (t,e), where $t \in T$ and $e = tv$ $(v \in V)$, i.e. e is an edge incident to t. If $\mathcal{P}$ is a set of paths, the union of the edge-transitions of the paths in $\mathcal{P}$ will be denoted by $\hat{\mathcal{P}}$. If $\mathcal{P}$ consists of edge-disjoint paths, each edge-transition is contained in at most one path. The following theorem and proof have been exposed in lectures [19], and appear in [18]. For the proof we introduce one more notation: the subpath of a path P between two of its points u, v is denoted by $P(u,v)$.

Theorem 1 *Let $G = (V,E)$ be a graph, $T \subseteq V$. Then $J_{\text{vertex}}(G,T), J_{\text{edge}}(G,T)$ are jump systems.*

Proof: As explained above, the proof for $J_{\text{vertex}}(G,T)$ and the one for $J_{\text{edge}}(G,T)$ can be obtained by mimicking one another: the most essential difference between the two is that the vertices in the former are becoming edges of the latter, and edges of the former become edge-transitions in the latter. We detail the full proof for $J_{\text{edge}}(G,T)$.

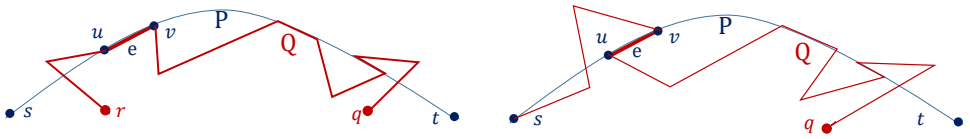


Figure 1: Explanation

Let m_1 and m_2 be two integer feasible vectors, and $\mathcal{P}_1, \mathcal{P}_2$ be a set of paths realizing them respectively. We use induction with respect to $|\hat{\mathcal{P}}_1 \setminus \hat{\mathcal{P}}_2|$ to prove that the 2-step axiom holds for m_1 and m_2.

Case 1: The first step is $-e_s$, where $s \in T$, $m_1(s) > m_2(s)$.

We show then the 2-step axiom for $-e_s$ as first step. If there exists an (s,t)-path $P \in \mathcal{P}_1$, so that $m_1(t) > m_2(t)$, then $-e_t$ is a correct second step that is realized by deleting P from $\mathcal{P}_1$. If such a path does not exist, we delete a path having s as an endpoint anyway, in the following way, keeping in mind that now $m_1(t) \leq m_2(t)$ holds for the other endpoint of such a path:

By $m_1(s) > m_2(s)$ there exists a path $P \in \mathcal{P}_1$ whose first edge is incident to s and is not contained in any path of $\mathcal{P}_2$. Therefore, deleting P from $\mathcal{P}_1$, $|\hat{\mathcal{P}}_1 \setminus \hat{\mathcal{P}}_2|$ decreases, enabling us to apply the 2-step axiom by induction. Recalling $m_1(t) \leq m_2(t)$, since after the deletion of the P, we get the feasible integer vector $m_1' = m_1 - e_s - e_t$, $m_1'(t) < m_2(t)$. Therefore we can apply the 2-step axiom: for e_t as first step there exists a feasible second step δ. In other words $m_1'' := m_1' + e_t + \delta = m_1 - e_s - e_t + e_t + \delta = m_1 - e_s + \delta$ is still a feasible integer vector, $m_1''(t) = m_1(t)$, so δ is the second step we were looking for.

Case 2: The first step is e_s, where $s \in T$, $m_1(s) < m_2(s)$.

We show then the 2-step axiom for e_s as first step. If there exists $t \in T$ along with an (s,t)-path $P \in \mathcal{P}_2$ such that P is edge-disjoint from all paths in $\mathcal{P}_1$, then

- either $m_1(t) < m_2(t)$ and then e_t is a correct second step realized by adding P to $\mathcal{P}_1$.
- or $m_1(t) \geq m_2(t)$, and then we show that there exists $Q \in \mathcal{P}_1$ with endpoint t, and that we can apply the induction hypothesis to $(\mathcal{P}_1 \setminus \{Q\}) \cup \{P\}$ to finish the proof.

Indeed, since $P \in \mathcal{P}_2$ is edge-disjoint from all paths in $\mathcal{P}_1$, strictly less than $m_1(t)$ edges incident to t are used by both $\mathcal{P}_1$ and $\mathcal{P}_2$, so there exists $Q \in \mathcal{P}_1$ whose edge incident to t is not used by any path of $\mathcal{P}_2$. The number of transitions of $(\mathcal{P}_1 \setminus \{Q\}) \cup \{P\}$ not contained in $\mathcal{P}_2$ is smaller than $|\hat{\mathcal{P}}_1 \setminus \hat{\mathcal{P}}_2|$, because P is in $\mathcal{P}_2$ so the union with P does not add anything, and by the choice of Q, the deletion of Q deletes at least one transition. Moreover, $\mathcal{P}_1 \setminus \{Q\} \cup \{P\}$ realizes the integer vector $m_1' := m_1 + e_s - e_q$, where q is the other endpoint of Q, so

- if $m_1(q) \leq m_2(q)$, then $m_1'(q) < m_2(q)$ and we can apply the induction hypothesis for m_1', m_2 and e_q as first step.

  With the second step δ, $m_1' + e_q + \delta = m_1 + e_s + \delta$ is then a feasible integer vector so δ is a good second step for m_1 and m_2 and e_s as first step,

- if $m_1(q) > m_2(q)$, then the feasibility of m_1' means that $-e_q$ is a good second step.

We finally suppose, still under the condition of Case 2, that there is no path with endpoint s in $\mathcal{P}_2$, which is edge-disjoint from all paths in $\mathcal{P}_1$. Since $m_1(s) < m_2(s)$, there exists $P \in \mathcal{P}_2$ with an edge incident to s not used by any path of $\mathcal{P}_1$. By our assumption, there exists an edge $e = uv$ of P also contained in a path $Q \in \mathcal{P}_1$, which is thus not incident to s, and suppose that starting from s on P, e is the first such edge we meet, and that we meet u before v. Thus neither u nor v are in T.

Let $q \in T$ be the endpoint of Q so that $Q(u,q)$ contains e (Figure 1 left), unless this endpoint is s, in which case define q to be the other endpoint of Q (Figure 1 right). So $q \neq s$ anyway, and $Q' := P(s,u) \cup Q(u,q)$ is a T-path disjoint from all paths of $\mathcal{P}_1 \setminus \{Q\}$. Now similarly to our repeated arguments, $(\mathcal{P}_1 \setminus \{Q\}) \cup \{Q'\}$ shows that $m_1' := m_1 + e_s - e_r$ is a feasible integer vector, where r is the other endpoint of Q. So $-e_r$ is a correct second step, provided $m_1(r) > m_2(r)$; on the other hand, if $m_1(r) \leq m_2(r)$, then $m_1'(r) < m_2(r)$, and we try to apply the 2-step axiom with e_r as first step. For this, it is sufficient to prove the Claim below, because that allows us to apply the induction hypothesis.

We will then be done, since for m_1', m_2, with a first step e_r and second step δ we have that $m_1 + e_s - e_r + e_r + \delta = m_1 + e_s + \delta$ is a feasible integer vector, finishing the verification of the 2-step axiom for Case 2, and therewith of our theorem. Indeed, then in our last case we can conclude the first step e_s with the second step δ. So the following Claim finishes the proof of the theorem:

Claim: $|(\hat{\mathcal{P}}_1 \setminus \hat{Q}) \cup \hat{Q}') \setminus \hat{\mathcal{P}}_2| < |\hat{\mathcal{P}}_1 \setminus \hat{\mathcal{P}}_2|$.

To prove this claim note first that all edge-transitions of the path $P(s,u)$ are contained both in Q' and $P \in \mathcal{P}_2$, so they are not counted in $|(\hat{\mathcal{P}}_1 \setminus \hat{Q}) \cup \hat{Q}') \setminus \hat{\mathcal{P}}_2|$; therefore there is no difference in the set of transitions of $P(s,u)$ and $Q(u,q) \subseteq Q$ between Q and Q'. Therefore it is sufficient to examine the transitions through vertex u.

Note that the edge-transitions through u decrease the induction parameter $|(\hat{\mathcal{P}}_1 \setminus \hat{Q}) \cup \hat{Q}') \setminus \hat{\mathcal{P}}_2|$ by 1 when Q is replaced by Q', if and only if Q' uses the same edge-transition in u as P, that is, if and only if $e \in Q'$ (Figure 1 left), and then we are done by the induction hypothesis. On the other hand, by our choice, $e \notin Q'$ happens only if $r = s$ (Figure 1 right), and then the edge-transition of Q in v, which contains e is counted in $|\hat{\mathcal{P}}_1 \setminus \hat{\mathcal{P}}_2|$ unless v is an endpoint of Q, and is no more contained $|(\hat{\mathcal{P}}_1 \setminus \hat{Q}) \cup \hat{Q}') \setminus \hat{\mathcal{P}}_2|$. But v is indeed, not an endpoint of Q, since e has been chosen not to be incident to s. □

3 Context and Consequences

The consequences of Mader's theorems [15], [16] for capacitated cases (that can be deduced by parallel edges or replications) are well-known, and Hu's [10], Rothchild and Whinston's [17], [12], ..., and also later generalizations concern arbitrary capacities. These raise new algorithmic questions though, beyond the size and ambitions of the present work. We therefore continue to restrict ourselves to the uncapacitated case knowing that from the viewpoint of theorems and structure the capacitated case is equivalent: for instance an integer edge-capacity can be simulated by the same number of parallel edges.

Let $G = (V, E)$ be a graph, and $T \subseteq V$. Definitions of feasible vectors replacing sets of openly vertex-disjoint T-paths by entirely vertex-disjoint paths joining different classes of a partition $\mathcal{T}$ of T are also easily seen to lead to equivalent feasibility problems. (In terms of combinatrial structure, while algorithmically, with binary encoding, this needs more explanations.) We will call such paths $\mathcal{T}$-paths. If a path is both a $\mathcal{T}_1$- and $\mathcal{T}_2$-path for partitions $\mathcal{T}_1, \mathcal{T}_2$ of T, it will be said to be a $\mathcal{T}_1$-$\mathcal{T}_2$-path. The set of vertex- or edge-*feasible vectors for* $\mathcal{T}_1 - \mathcal{T}_2$ is defined analogously to that for $\mathcal{T}$, leading to the study of some particular jump system intersections. Vertex-feasible vectors for $\mathcal{T}$ or for $\mathcal{T}_1 - \mathcal{T}_2$ are $0-1$ vectors and Schrijver proved that the maximal ones among them form the bases of a matroid; Theorem 1 sharpens this to the fact that the vertex-feasible vectors for $\mathcal{T}$ form a delta-matroid. In the rest of this article we focus on edge-feasible vectors.

We can also define *relaxed-feasibility* for $\mathcal{T}$-paths or for $\mathcal{T}_1$-$\mathcal{T}_2$-paths by extending the definition to not necessarily integer vectors in $\mathbb{R}^T$ with the existence of (not necessarily integer) coefficients for each path so that the sum of coefficients containing each $e \in E$ is at most 1. We state here some preliminaries to Theorem 1 from [19], [18] without proof details, but point at connections and open problems related

to these. The presentation of jump-systems by these early results is weaker than Theorem 1: taking all integer vectors in the convex hull of $J_{\text{edge}}(G,T)$, or the assumption of a parity condition are essential weakenings. However, further facts, an intersection theorem and an intriguing conjecture can be exhibited for these restricted jump systems, with interesting, also algorithmic consequences on edge-disjoint T-paths.

Theorem 2 *Given the graph $G=(V,E)$, $T\subseteq V$, and a partition $\mathcal{T}$ of T, the vertices of the polytope*

$$Q(G,\mathcal{T}) := \{m\in {\mathbb{R}_+}^T : m(X\cap T') - m(X\cap T\setminus T') \le d(X) \text{ for all } X\subseteq V,\ T'\in\mathcal{T}\}$$

are integer, its integer points form a jump sytem, and $Q(G,\mathcal{T})$ is the set of relaxed-feasible vectors.

It is easy to check that the inequalities defining $Q(G,\mathcal{T})$ are satisfied by any relaxed-feasible vector.

Attention! The encouraging facts stated in the theorem do not imply that integer points in $Q(G,\mathcal{T})$ are feasible. Actually not all of them are, even though the membership oracle for $J_{\text{edge}}(G,T)$ can be straightforwardly reduced to Mader's theorem. The situation is more difficult for the intersection of such jump systems, as we try to show with the following results and conjecture.

The linear inequalities describing $Q(G,\mathcal{T})$ are from [7, Theorem 6.1], where it is also proved that the integer vectors $m\in Q(G,\mathcal{T})$ for which $m+d_G$ is even (in this sum m is defined to be 0 on $V\setminus T$) for all $v\in V$, are feasible. Furthermore, the same is true for $m\in Q(G,\mathcal{T}_1)\cap Q(G,\mathcal{T}_2)$, where $\mathcal{T}_1$, $\mathcal{T}_2$ are two partitions of T, (and actually even more generally). This is in the line of multiflow maximization results of Hu [10], Rothschild and Whinston [17], Cherkasskiĭ and Lovász [4], [13], Karzanov and Lomonosov [12] *under parity constraints on the degrees*, and if the parity constraint is not supposed, only a half-integer solution can be stated. (Such a half-integer solution thus exists for all relaxed feasible vectors.) A merit of [7] is to introduce vectors on T ("node-demands") as an intermediate tool. Then the goal of maximization can be achieved using matroid intersection, implying the minmax theorems corresponding to all the mentioned results.

A *bisubmodular polyhedron* is a polyhedron of the form

$$Q(b) := \{x\in\mathbb{R} :\ x(A)-x(B)\le b(A,B),\ x\ge 0\},$$

where b is defined on pairs of disjoint sets and has values in $\mathbb{N}$, moreover it is *bisubmodular*, that is:

$$f(A,B)+f(A',B')\ge f(A\cap A',B\cap B')+f((A\cup A')\setminus(B\cup B'),(B\cup B')\setminus(A\cup A')).$$

Denote by $\text{conv}(X)$ the convex hull of the set $X\subseteq\mathbb{R}^n$.

Corollary 3 *If G is an arbitrary unirected graph and $T\subseteq V$, $\text{conv}(J_{\text{vertex}}(G,T))$ and $\text{conv}(J_{\text{edge}}(G,T))$ are bisubmodular polyhedra.*

This shows one of the utilities of J being a jump system, having the consequence that linear objective function can be optimized with a natural greedy algorithm in an appropriate oracle context [2] satisfied by the combinatorial examples we know about.

Proof: By Theorem 1 $J_{\text{vertex}}(G,T)$ and $J_{\text{edge}}(G,T)$ are jump systems, and Bouchet, Cunningham [2] proved that the convex hull of each jump system is a bisubmodular polyhedron. □

Theorem 4 *$Q(G,\mathcal{T})$ is a bisubmodular polyhedron. If all degrees of G are even, then for any two partitions $\mathcal{T}_1$, $\mathcal{T}_2$ of T, $Q(G,\mathcal{T}_1)\cap Q(G,\mathcal{T}_2)$ has integer vertices, and the maximum of the sum of coordinates on this intersection is achieved on a feasible vector computable in polynomial time.*

Proof: For checking fact that $Q(G,\mathcal{T})$ is a bisubmodular polyhedron note first that it is defined by a ±1 constraint matrix. Then the bisubmodular inequality can be verified directly [19], we omit the details here, they are included in [18].

If all degrees are even, the bisubmodular function defining $Q(G, \mathcal{T})$ has only even values. According to Cunningham [5] *the intersection of bisubmodular polyhedra is half-integer.* The original proof of this used polyhedral arguments, and was not more difficult than apparently the only proof that appeared publicly, in [24]. It is deduced there from a more general conjecture of Cunningham for jump systems, and then Cunningham's conjecture is settled using results in [14].

If all degrees of G are even, then both bisubmodular functions b_1 and b_2 defining $\mathcal{T}_1$ and $\mathcal{T}_2$ are even, so $b_i/2$ $(i = 1, 2)$ are integer bisubmodular functions. Applying the already established half-integrality for the intersection of the polytopes defined by $b_i/2$ $(i = 1, 2)$, we get that $Q(G, \mathcal{T}_1) \cap Q(G, \mathcal{T}_2)$ is an integer polyhedron. The last statement neecessitates a completely different proof method, it is proved in [7, page 165 Proof of Theorem 4.3]. □

The proof of the first statement of Theorem 4 is easier than that of Theorem 1, which has been proved almost a year later. The latter does not easily imply though the former: integer vectors of $Q(G, \mathcal{T})$ are not necessarily edge-feasible without the parity condition (Mader's odd cuts also play then a role for feasibility). Accordingly, integer vectors of $Q(G, \mathcal{T})$ are not necessarily equal to $\mathrm{conv}(J_{\mathrm{edge}}(G, T))$.

However, if the degree of every vertex of G is even, the bisubmodular functions defining $Q(G, \mathcal{T})$ are even, the optimum m on $Q(G, \mathcal{T})$ is an even vector for any objective function, and then [7, Theorem 6.1] explicitly establishes the feasibility of m. It actually does so for all even vectors $m \in Q(G, \mathcal{T}_1) \cap Q(G, \mathcal{T}_2)$. (For the $\mathcal{T} = \mathcal{T}_1 = \mathcal{T}_2$ special case that we need here, an easy reduction to [13], [4] is actually sufficient and can be realized by adding a copy t' of each terminal, and $m(t)$ paralel tt' edges. The same reduction works for testing membership in the jump systems $J_{\mathrm{edge}}(G, T)$, $J_{\mathrm{vertex}}(G, T)$ in polynomial time using any algorithm for maximizing T-paths.)

For $\mathcal{T}_1 - \mathcal{T}_2$-paths in graphs whose degrees are not all even, we would need to generalize Mader's theorem to $\mathcal{T}_1 - \mathcal{T}_2$-paths, and such a generalization does not exist since the 2-flow problem is already $\mathcal{NP}$-hard. Now $m \in Q(G, \mathcal{T}_1) \cap Q(G, \mathcal{T}_2)$ is not even any more so [7] cannot be applied (see [7] for further explanations and examples). Nevertheless, for maximixing the sum of coordinates [7] presents a patch, expressed in the last statement of Theorem 4, and suggesting that the same may also be true for *all vertices* of the polytope $Q(G, \mathcal{T}_1) \cap Q(G, \mathcal{T}_2)$, making possible the optimization of any objective function on $\mathcal{T}_1 - \mathcal{T}_2$-feasible vectors if the following conjecture holds:

Conjecture 5 *If all degrees of G are even, then for any two partitions $\mathcal{T}_1$, $\mathcal{T}_2$ of T, the vertices of $Q(G, \mathcal{T}_1) \cap Q(G, \mathcal{T}_2)$ are $\mathcal{T}_1 - \mathcal{T}_2$-feasible, i.e. $Q(G, \mathcal{T}_1) \cap Q(G, \mathcal{T}_2)$ is the convex hull of $\mathcal{T}_1 - \mathcal{T}_2$-feasible vectors.*

If true, the assertion of this conjecture would immediately imply the polynomial computability of optimal feasible vectors for arbitrary weights on T. Surprisingly, this seems to be doable, without knowing whether the conjecture is true, in a general context containing both graph factors and paths (see below).

Another use of knowing that a set is a jump-system is that one can sometimes decide feasibility or optimize on the intersection with some other jump-systems. A recent breakthrough by Dudycz and Paluch's [6] on graph factors has been simplified and extended by Kobayashi [11] to the abstract level of jump systems. Feasible sets defined above from disjoint paths problems can be easily proved to satisfy Kobayashi's conditions – by pursuing Schrijver's reduction [21] to Gallai's theorem [8] where for the appropriate generalization "general factors" may be used instead of matchings – to conclude with weighted optimization without confirming Conjecture 5.

Kobayashi's conditions [11, Theorem 5.1] can potentially handle multiflow theorems with various kinds of weightings. For jump systems in general, the simple algorithm reducing the feasibility of general factors of graphs to parity constrained factors [22] (explained briefly in [3]), works for the intersection of some jump systems [23], making possible to compute the oracle required in [11, C_1', C_3'], and enabling the use of [11, Theorem 1.4], under generalized conditions.

Results on deciding the emptiness or finding an element of some jump-system-intersections are being explored in more details in a forthcoming article.

Acknowledgment: The authors are indebted to Satoru Iwata and Yu Yokoi for several relevant correction/update turns.

References

[1] A. Bouchet, Greedy algorithm and symmetric matroids *Mathematical Programming*, **38** (1987), pp. 147–159.

[2] A. Bouchet and W. Cunningham, Delta-matroids, Jump-systems and Bisubmodular polyhedra, *SIAM J. Discrete Mathematics*, Series **8** (1995), pp. 17-32.

[3] G. Cornuéjols, General Factors of Graphs, Journal of Combinatorial Theory, B **45** (1988), pp. 185–198.

[4] B. V. Cherkasskiĭ, A solution of a problem of multicommodity flows in a network, Èkonomika i Matematicheskie Metody, **13**, 1977, 143–151

[5] W. H. Cunningham, *private communication* (1994), a proof from a generalization appeared in [24].

[6] S. Dudycz, K. Paluch, Optimal General Matchings, (May 2021)
`http://arxiv.org/abs/1706.07418`

[7] A. Frank, A. Karzanov, A. Sebő, On integer multiflow maximisation, *SIAM J. Discrete Mathematics*, **10 (1)**, (1997), 158–170.

[8] T. Gallai, Maximum-minimum Sätze und verallgemeinerte Faktoren von Graphen, *Acta Mathematica Academiae Scientiarum Hungaricae* **12** (1961), 131–173.

[9] S. Iwata, Y. Yokoi, Openly Disjoint Paths, Jump Systems, and Discrete Convexity, *Proceedings of the 12th Japanese-Hungarian Symposium on Discrete Mathematics and Its Applications*, 2023.

[10] T. C. Hu, Multi-commodity network flows,, *Operations Research*, **9**, (1963), 344–360.

[11] Y. Kobayashi , Optimal General Factor Problem and Jump System Intersection,
`https://arxiv.org/abs/2209.00779`, October 2022.

[12] M.V. Lomonosov, Combinatorial Approaches to Multiflow Problems, *Discrete Applied Mathematics*, **11, (1)** (1985), 1–93

[13] L. Lovász, On some connectivity properties of Eulerian graphs, *Acta Acad. Sci. Hung.*, **28** (1976), 129–138.

[14] L. Lovász, The Membership Problem in Jump Systems, *Journal of Combinatorial Theory*, Series **B, 70** (1997), 45–66.

[15] W. Mader Über die Maximalzahl kantendisjunkter A-Wege, *Arch. Math.*, **30** (1978) 325–336.

[16] W. Mader Über die Maximalzahl kreuzungsfreier H-Wege, *Arch. Math.*, **31** (1978) 387–402

[17] B. Rothschild and A.Whinston Feasibility of two-commodity network flows, *Oper. Res.* **14** (1966) 1121–1129.

[18] M. Sadli, Généralisations de matroïdes et chemins disjoints, thèse pour obtenir le grade de Docteur de L'Institut National Polytechnique de Grenoble, June 26, 2000 (in French).

[19] M. Sadli, A. Sebő, Paths and Jumps, *Notes and lectures,* (1999-2000), Grenoble–Waterloo–Alpes d'Huez–Meylan.

[20] A. Schrijver. *Combinatorial Optimization*, *Springer-Verlag Berlin Heidelberg*, 2003.

[21] A. Schrijver A short proof of Mader's $\mathcal{S}$-paths theorem, *Journal of Combinatorial Theory*, Series B,**82** (2001), 319–321.

[22] A. Sebő, Gráfok faktorai: struktúrák és algoritmusok, *kandidátusi értekezés,* 1987 augusztus.

[23] A. Sebő,General factors and Jump System Intersections, *Lecture at Workshop on "Matroids, Matchings and Extensions",* (December 1999), Special Year on Graph Theory and Combinatorial Optimization.

`http://www.fields.utoronto.ca/programs/scientific/99-00/graph_theory/matroids_matching/`

[24] A. Sebő, Gaps and Jumps, *Third Annual DONET meeting, 1996)*, M. Klazar ed.

`https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=ee2689dacab3d9c89806eb84827d9560dae341a9`

Characterization and Algorithm for Bivariate Multi-Unit Assignment Valuations

Takafumi Otsuka

Department of Industrial Engineering and Economics,
Tokyo Institute of Technology,
Tokyo 152-8550, Japan

Akiyoshi Shioura[2]

Department of Industrial Engineering and Economics,
Tokyo Institute of Technology,
Tokyo 152-8550, Japan
shioura.a.aa@m.titech.ac.jp

Abstract: A multi-unit assignment valuation is a function represented by a weighted bipartite graph. In this paper, we provide a characterization of such a function in terms of maximizer sets of purturbed functions. We then present an algorithm that checks whether a given bivariate function is a multi-unit assignment valuation, and if the answer is "yes," computes a weighted bipartite graph representing the function.

Keywords: algorithm, discrete convex function, auction, assignment valuation

1 Introduction

A valuation function is a function that for a given set of goods, returns the value of the set. This paper deals with multi-unit valuation functions defined on non-negative integral vectors $\mathbb{Z}_+^n$, in which a vector $x \in \mathbb{Z}_+^n$ represents a multiset of n discrete goods. We consider a class of multi-unit valuation functions that are represented by weighted bipartite graphs, which are referred to as *multi-unit assignment valuation functions*.

Given a complete bipartite graph $G = (V, N; V \times N)$ with a weight function $w : V \times N \to \mathbb{R}$ and a supply function $\varphi : V \to \mathbb{Z}_{++}$, a multi-unit assignment valuation function $f : T_\Phi \to \mathbb{R}$ is defined by

$$f(x) = \max \left\{ \sum_{(i,j) \in V \times N} w(i,j) y(i,j) \;\middle|\; \sum_{i \in V} y(i,j) = x(j) \ (j \in N), \ \sum_{j \in N} y(i,j) \le \varphi(i) \ (i \in V), \right.$$
$$\left. y(i,j) \in \mathbb{Z}_+ \ (i \in V, j \in N) \right\} \qquad (x \in T_\Phi),$$

where

$$N = \{1, 2, \ldots, n\}, \qquad \Phi = \sum_{i \in V} \varphi(i), \qquad T_\Phi = \{x \in \mathbb{Z}_+^n \;\middle|\; \sum_{j \in N} x(j) \le \Phi\}.$$

In the case where the effective domain T_Φ is restricted to zero-one vectors, function f is nothing but an assignment valuation [16], which often appears in the literature of auction theory. In the following, we simply refer to function f as an *assignment valuation* when no confusion arises.

It is known that the class of assignment valuations is a proper subclass of strong-substitutes valuations (see, e.g., [13, 14, 17]). The strong-substitutes condition for a multi-unit valuation [10] is a natural generalization of the gross-substitutes condition for a single-unit valuation due to Kelso and Crawford [8] (see also Gul and Stacchetti [6, 7]), and the former condition inherits various nice properties of the latter condition. In particular, strong-substitutes condition for bidders' valuations implies the existence

[2]This work is supported by JSPS KAKENHI Grant Numbers 18K11177.

of Walrasian equilibrium in the auction market with multiple units of indivisible goods. Also, strong-substitutes valuations are known to be equivalent to $\mathrm{M}^\natural$-concave functions in discrete convex analysis [4, 15] (see also [13, 14, 17]). This fact implies that an assignment valuation also enjoys nice properties as a discrete concave function.

We are interested in a special case of assignment valuations with $N = \{1, 2\}$, motivated by the "product-mix" auction used in Bank of England [9]. The auction in Bank of England deals with two kinds of multiple discrete goods. Each bidder of the auction expresses its demand to the goods by using a set of "weighted bid vectors;" a weighted bid vector is a pair (b, ω) of a bidding price vector $b \in \mathbb{R}^2$ and its weight $\omega \in \mathbb{Z}_{++}$. It turns out that sets of weighted bid vectors have a natural one-to-one correspondence with assignment valuations, and the demand information represented by a set of weighted bid vectors is the same as the one represented by the corresponding assignment valuation; see Remark 4.1 in Section 3 for more details on this relationship.

Suppose that a bidder wants to participate the product-mix auction with its own valuation function. In such a situation a bidder wants to know whether its valuation can be represented as an assignment valuation, and if it is an assignment valuation, the bidder also wants to know the representation by a weighted bipartite graph. This motivates us to consider the following Bivariate Assignment Valuation Checking Problem (BAVCP):

> given a bivariate valuation function $f : T_\Phi \to \mathbb{R}$ with some positive integer Φ, which is not necessarily an assignment valuation, answer whether f is an assignment valuation, and if the answer is "yes," then find a weighted bipartite graph representing f.

Our goal in this paper is to propose an efficient algorithm for solving this problem.

To develop an algorithm for the problem (BAVCP), we first provide a characterization of assignment valuations in terms of maximizer sets. For a bivariate function $f : T_\Phi \to \mathbb{R}$ and a vector $p \in \mathbb{R}^2$, we define a set

$$D_f(p) = \{x \in T_\Phi \mid f(x) - p^\top x \geq f(y) - p^\top y \ (y \in T_\Phi)\},$$

which is called a *maximizer set* (also called a *demand set*). As mentioned above, every assignment valuation is an $\mathrm{M}^\natural$-concave function, for which the following characterization in terms of maximizer sets is known; definitions of $\mathrm{M}^\natural$-concave function and $\mathrm{M}^\natural$-convex set will be given in Section 2.

Theorem 1.1 (cf. [11, 12])**.** *A bivariate function $f : T_\Phi \to \mathbb{R}_+$ is $\mathrm{M}^\natural$-concave if and only if for every $p \in \mathbb{R}^2$ the maximizer set $D_f(p)$ is an $\mathrm{M}^\natural$-convex set.*

Since an assignment valuation is an $\mathrm{M}^\natural$-concave function, its maximizer set is an $\mathrm{M}^\natural$-convex set, which is (the set of integral vectors in) a hexagon. We classify $\mathrm{M}^\natural$-convex sets into three types based on the length of six edges: positive-type, zero-type, and negative-type (see Section 2.2 for precise definitions), and show that an assignment valuation can be characterized by a stronger property for maximizer sets.

Theorem 1.2. *A bivariate function $f : T_\Phi \to \mathbb{R}$ with $f(0, 0) = 0$ is an assignment valuation if and only if for every $p \in \mathbb{R}^2$ the maximizer set $D_f(p)$ is an $\mathrm{M}^\natural$-convex set of positive-type or zero-type.*

We will also show that for an assignment valuation f, the information about its representation can be obtained from maximizer sets of f that are $\mathrm{M}^\natural$-convex sets of positive-type. Based on the results, we propose an algorithm for the problem (BAVCP) that runs in $\mathrm{O}(\Phi^2)$ time, under the assumption that the value oracle for f is available; given a vector x, the value oracle returns the value $f(x)$.

Finally, we note that a closely related problem is discussed by Goldberg, Lock, and Marmolejo-Cossío [5]. In our terminology, their problem is described as follows: the input is an n-variate multi-unit assignment valuation $f : T_\Phi \to \mathbb{R}$, represented by a *demand oracle*, and the output is a weighted bipartite graph representing f; given a vector $p \in \mathbb{R}^n$, the demand oracle returns a vector in the maximizer set $D_f(p)$. The algorithm proposed in [5] runs in $\mathrm{O}(n|V| \log W)$ time with $W = \max\{w(i, j) \mid i \in V,\ j \in N\}$. It is known (see, e.g., [13]) that a demand oracle can be realized by using a value oracle in $\mathrm{O}(n^3 \log(\Phi/n))$ time. Hence, if the value oracle is available, then the algorithm in [5] runs in $\mathrm{O}(n^4|V| \log W \log(\Phi/n))$ time. In particular, if $n = 2$ (i.e., f is a bivariate function), then the algorithm runs in $\mathrm{O}(|V| \log W \log \Phi)$

time, which is incomparable to the running time $O(\Phi^2)$ of our algorithm since $|V| \leq \Phi$ and the parameter W does not appear in ours. Also, it should be noted that our algorithm checks whether a given function is an assignment valuation or not, while the algorithm in [5] does not.

2 Preliminaries

2.1 Multi-unit Assignment Valuation and M$^\natural$-concave Function

A bivariate assignment valuation $f : T_\Phi \to \mathbb{R}$ is defined as follows by using a complete bipartite graph $G = (V, \{1,2\}; V \times \{1,2\})$ with weight function $w : V \times \{1,2\} \to \mathbb{R}$ and supply function $s : V \to \mathbb{Z}_{++}$:

$$f(x) = \max \left\{ \sum_{i \in V} (w(i,1)y(i,1) + w(i,2)y(i,2)) \;\middle|\; \sum_{i \in V} y(i,j) = x(j) \ (j = 1,2), \right.$$
$$y(i,1) + y(i,2) \leq \varphi(i) \ (i \in V),$$
$$\left. y(i,j) \in \mathbb{Z}_+ \ (i \in V, \ j = 1,2) \right\} \quad (x \in T_\Phi), \tag{2.1}$$

where $\Phi = \sum_{i \in V} \varphi(i)$ and $T_\Phi = \{x \in \mathbb{Z}_+^2 \mid x(1) + x(2) \leq \Phi\}$. We may assume, without loss of generality, that

$$\forall i, i' \in V \text{ with } i \neq i', \ w(i,1) \neq (i',1) \text{ or } w(i,2) \neq w(i',2) \text{ (or both)}; \tag{2.2}$$

if there exist distinct $i, i' \in V$ with $w(i,1) = (i',1)$ and $w(i,2) = w(i',2)$, then we can replace $\varphi(i)$ with $\varphi(i) + \varphi(i')$ and delete the vertex i', which results in the same assignment valuation.

It is known that a (not necessarily bivariate) assignment valuation has a nice discrete structure called M$^\natural$-concavity (see, e,g., [13]).

Proposition 2.1. *A (bivariate) assignment valuation is an M$^\natural$-concave function.*

A bivariate function $f : T_\Phi \to \mathbb{R}$ is said to be *M$^\natural$-concave* if it satisfies the following exchange property for every $x, y \in T_\Phi$:

(M$^\natural$-EXC) for each $i \in N = \{1,2\}$ with $x(i) > y(i)$, we have either (i), (ii), or both:
(i) $y(N) < \Phi$ and $f(x) + f(y) \leq f(x - \chi_i) + f(y + \chi_i)$,
(ii) there exists some $i' \in N$ with $x(i') < y(i')$ such that

$$f(x) + f(y) \leq f(x - \chi_i + \chi_{i'}) + f(y + \chi_i - \chi_{i'}),$$

where $\chi_i \in \{0,1\}^N$ denotes the characteristic vector of $i \in N$, i.e., $\chi_1 = (1,0)$ and $\chi_2 = (0,1)$. Note that $x - \chi_i, y + \chi_i \in T_\Phi$ holds in the case of (i), and $x - \chi_i + \chi_{i'}, \ y + \chi_i - \chi_{i'} \in T_\Phi$ holds in the case of (ii).

M$^\natural$-concave functions can be characterized by a local exchange property: f is M$^\natural$-concave if and only if (M$^\natural$-EXC) holds for every $x, y \in T_\Phi$ with $\|x - y\|_1 \leq 4$. This local exchange property can be specialized for bivariate functions f as follows:

$$f(k,h) + f(k+1,h+1) \leq f(k+1,h) + f(k,h+1) \quad ((k,h) \in \mathbb{Z}_+^2, \ k + h + 2 \leq \Phi), \tag{2.3}$$
$$f(k,h+1) + f(k+2,h) \leq f(k+1,h+1) + f(k+1,h) \quad ((k,h) \in \mathbb{Z}_+^2, \ k + h + 2 \leq \Phi), \tag{2.4}$$
$$f(k+1,h) + f(k,h+2) \leq f(k+1,h+1) + f(k,h+1) \quad ((k,h) \in \mathbb{Z}_+^2, \ k + h + 2 \leq \Phi). \tag{2.5}$$

Proposition 2.2. *A bivariate function $f : T_\Phi \to \mathbb{R}$ is M$^\natural$-concave if and only if it satisfies the conditions* (2.3), (2.4), *and* (2.5).

The conditions (2.3), (2.4), and (2.5) can be understood in terms of "triangles." For $k, h \in \mathbb{Z}$, we define an *upper-right triangle* $T_{\rm ur}(k,h) \subseteq \mathbb{Z}^2$ and a *lower-left triangle* $T_{\rm ll}(k,h) \subseteq \mathbb{Z}^2$ by

$$T_{\rm ur}(k,h) = \{(k,h),(k-1,h),(k,h-1)\}, \qquad T_{\rm ll}(k,h) = \{(k,h),(k+1,h),(k,h+1)\}.$$

The condition (2.3) means that the function f bends upward on $T_{\rm ur}(k+1,h+1) \cup T_{\rm ll}(k,h)$. Similarly, (2.4) (resp., (2.5)) means that that f bends upward on $T_{\rm ur}(k+1,h+1) \cup T_{\rm ll}(k,h+1)$ (resp. $T_{\rm ur}(k+1,h+1) \cup T_{\rm ll}(k+1,h)$).

2.2 M$^\natural$-convex Set and Its Properties

We also define M$^\natural$-convexity for a set $S \subseteq \mathbb{Z}^2$ as follows. For a non-empty set $S \subseteq \mathbb{Z}^2$, we say that S is an *M$^\natural$-convex set* if it satisfies the following exchange property for every $x, y \in S$:

for each $i \in N = \{1,2\}$ with $x(i) > y(i)$, at least one of (i) and (ii) holds:
(i) $x - \chi_i, y + \chi_i \in S$, (ii) $x - \chi_i + \chi_{i'},\ y + \chi_i - \chi_{i'} \in S$ for some $i' \in N$ with $x(i') < y(i')$.

We present some properties on polyhedral structure of M$^\natural$-convex sets in $\mathbb{Z}^2$. M$^\natural$-convex sets can be described by simple inequalities.

Proposition 2.3. *A bounded set $S \subseteq \mathbb{Z}^2$ is an M$^\natural$-convex set if and only if it can be represented by the following system of inequalities:*

$$S = \{(x(1),x(2)) \in \mathbb{Z}^2 \mid \lambda_1 \le x(1) \le \mu_1,\ \lambda_2 \le x(2) \le \mu_2,\ \lambda_0 \le x(1)+x(2) \le \mu_0\}. \tag{2.6}$$

We may assume that all inequalities in (2.6) are tight, i.e., it holds that

$$\lambda_i = \min\{x(i) \mid x \in S\}, \quad \mu_i = \max\{x(i) \mid x \in S\} \quad (i = 1,2),$$
$$\lambda_0 = \min\{x(1)+x(2) \mid x \in S\}, \quad \mu_0 = \max\{x(1)+x(2) \mid x \in S\}.$$

We see from the representation (2.6) that every bounded two-dimensional (2-d, for short) M$^\natural$-convex set S can be represented as a union of upper-right triangles $T_{\rm ur}(k,h)$ and lower-left triangles $T_{\rm ll}(k,h)$.

Let $S \subseteq \mathbb{Z}^2$ be a bounded 2-d M$^\natural$-convex set. We denote by $\overline{S} \subseteq \mathbb{R}^2$ the convex hull of S. Proposition 2.3 implies that the convex hull $\overline{S}$ is represented by the same set of inequalities in (2.6), and the six vertices of $\overline{S}$ are given (in clockwise order) as

$$(\lambda_1,\mu_2),\ (\mu_0-\mu_2,\mu_2),\ (\mu_1,\mu_0-\mu_1),\ (\mu_1,\lambda_2),\ (\lambda_0-\lambda_2,\lambda_2),\ (\lambda_1,\lambda_0-\lambda_1); \tag{2.7}$$

all of these vertices are integral and therefore contained in S. We also have $\overline{S} \cap \mathbb{Z}^2 = S$.

This observation shows that an M$^\natural$-convex set S can be identified with its convex hull $\overline{S}$. Hence, we can define an *edge* of S as the set of integral vectors in an edge of $\overline{S}$. In particular, the convex hull $\overline{S}$ is a hexagon with six edges, and therefore we can define *upper-horizontal (UH) edge, lower-horizontal (LH) edge, left-vertical (LV) edge, right-vertical (RV) edge, upper-right-diagonal (URD) edge,* and *lower-left-diagonal (LLD) edge.*

We also define the length of an edge in an M$^\natural$-convex set S by the length of the corresponding edge in the convex hull $\overline{S}$. We denote by

$$\ell_{\rm UH}(S), \quad \ell_{\rm LH}(S), \quad \ell_{\rm LV}(S), \quad \ell_{\rm RV}(S), \quad \ell_{\rm URD}(S), \quad \ell_{\rm LLD}(S)$$

the length of UH-edge, LH-edge, LV-edge, RV-edge, URD-edge, and LLD-edge; it is possible that some edges may have length zero.

By using six vertices in (2.7), the length of six edges are given as

$$\left.\begin{array}{ll} \ell_{\rm UH}(S) = (\mu_0-\mu_2)-\lambda_1, & \ell_{\rm LH}(S) = \mu_1-(\lambda_0-\lambda_2), \\ \ell_{\rm LV}(S) = \mu_2-(\lambda_0-\lambda_1), & \ell_{\rm RV}(S) = (\mu_0-\mu_1)-\lambda_2, \\ \ell_{\rm URD}(S) = \sqrt{2}(\mu_1-(\mu_0-\mu_2)) = \sqrt{2}(\mu_2-(\mu_0-\mu_1)), & \\ \ell_{\rm LLD}(S) = \sqrt{2}((\lambda_0-\lambda_2)-\lambda_1) = \sqrt{2}((\lambda_0-\lambda_1)-\lambda_2). & \end{array}\right\} \tag{2.8}$$

This immediately implies the following relations for the six edge lengths.

Proposition 2.4. *For a two-dimensional $M^\natural$-convex set $D \subseteq \mathbb{Z}^2$, it holds that*

$$\ell_{\mathrm{LH}}(D) - \ell_{\mathrm{UH}}(D) = \ell_{\mathrm{LV}}(D) - \ell_{\mathrm{RV}}(D) = (\ell_{\mathrm{URD}}(D) - \ell_{\mathrm{LLD}}(D))/\sqrt{2}.$$

Using the edge length, we classify bounded 2-d $\mathrm{M}^\natural$-convex sets. We say that a bounded $\mathrm{M}^\natural$-convex set $D \subseteq \mathbb{Z}^2$ is *positive-type* (resp., *negative-type*) if $\ell_{\mathrm{LH}}(D) - \ell_{\mathrm{UH}}(D) > 0$ (resp. $\ell_{\mathrm{LH}}(D) - \ell_{\mathrm{UH}}(D) < 0$), and *zero-type* otherwise (i.e., D is not two-dimensional or satisfies $\ell_{\mathrm{LH}}(D) - \ell_{\mathrm{UH}}(D) = 0$).

2.3 Properties of $\mathrm{M}^\natural$-concave Functions

We present some properties of $\mathrm{M}^\natural$-concave functions used in this paper. See [13] for more accounts on $\mathrm{M}^\natural$-concave functions.

For a bivariate function $f : T_\Phi \to \mathbb{R}$ and a vector $p \in \mathbb{R}^2$, the *maximizer set* $D_f(p) \subseteq T_\Phi$ is defined as

$$D_f(p) = \{x \in T_\Phi \mid f(x) - p^\top x \geq f(y) - p^\top y \ (y \in T_\Phi)\};$$

$D_f(p)$ is often referred to as a *demand set* in the context of auction, where f is regarded as a valuation for multisets of goods. If $D \subseteq T_\Phi$ is a two-dimensional maximizer set, then there exists a unique $p \in \mathbb{R}^2$ such that $D = D_f(p)$; we call such p the *slope vector* of D.

$\mathrm{M}^\natural$-concavity of a function can be characterized in terms of maximizer sets.

Proposition 2.5. *A bivariate function $f : T_\Phi \to \mathbb{R}$ is $M^\natural$-concave if and only if $D_f(p)$ is an $M^\natural$-convex set for every $p \in \mathbb{R}^2$.*

An $\mathrm{M}^\natural$-concave function can be extended to a polyhedral concave function. For a function $f : T_\Phi \to \mathbb{R}$, the *concave closure* $\bar{f} : \overline{T_\Phi} \to \mathbb{R}$ is defined as

$$\bar{f}(y) = \inf\{p^\top y + \eta \mid p \in \mathbb{R}^2,\ \eta \in \mathbb{R},\ p^\top x + \eta \geq f(x)\ (x \in T_\Phi)\} \qquad (y \in \overline{T_\Phi}).$$

By definition, $\bar{f}$ is a polyhedral concave function satisfying $\bar{f}(x) \geq f(x)$ for all $x \in T_\Phi$.

Proposition 2.6. *For an $M^\natural$-concave function $f : T_\Phi \to \mathbb{R}$, $\bar{f}(x) = f(x)$ holds for every $x \in T_\Phi$.*

Let $f : T_\Phi \to \mathbb{R}$ be a bivariate $\mathrm{M}^\natural$-concave function. It follows from Proposition 2.6 that for every $p \in \mathbb{R}^2$, the convex hull $\overline{D_f(p)}$ of the maximizer set of f coincides with a maximizer set $\{x \in \overline{T_\Phi} \mid \bar{f}(x) - p^\top x \geq \bar{f}(y) - p^\top y \ (y \in \overline{T_\Phi})\}$ of the concave closure $\bar{f}$. It is also known that a polyhedral subdivision of the polytope $\overline{T_\Phi}$ can be obtained from the family $\{\overline{D_f(p)} \mid p \in \mathbb{R}^2\}$ of convex hulls of maximizer sets. These facts imply that the information about the set of 2-d maximizer sets uniquely determines the function values of f. For two 2-d maximizer sets D and D' of f, we say that D and D' are *adjacent* if they share an edge of positive length.

Proposition 2.7. *Let $f : T_\Phi \to \mathbb{R}$ be a bivariate $M^\natural$-concave function with $f(0,0) = 0$. Then, the function values $f(x)$ $(x \in T_\Phi)$ are uniquely determined by the following information:*

- *the family $\mathcal{D} = \{D_i \mid i \in V\}$ of two-dimensional maximizer sets of f, where V is an appropriately chosen index set.*
- *the adjacency relation among maximizer sets in $\mathcal{D}$,*
- *the slope vector $p_i \in \mathbb{R}^2$ of D_i for $i \in V$,*
- *lengths $\ell_{\mathrm{LH}}(D_i), \ell_{\mathrm{UH}}(D_i), \ell_{\mathrm{LV}}(D_i), \ell_{\mathrm{RV}}(D_i), \ell_{\mathrm{URD}}(D_i), \ell_{\mathrm{LLD}}(D_i)$ of six edges for $i \in V$.*

3 Characterization of Multi-Unit Assignment Valuations and Algorithm

Main results of this paper are presented in this section. We first provide a characterization of bivariate assignment valuations by using the following condition for maximizer sets:

(MS$\geq$) every maximizer set of f is an M$^\natural$-convex set of positive-type or zero-type;

recall that every maximizer set of an M$^\natural$-concave function f is an M$^\natural$-convex set. In the following, we may simply say that a maximizer set of a bivariate M$^\natural$-concave function is of *positive-type* (resp. *zero-type*) if it is a M$^\natural$-convex set of positive-type (resp., zero-type).

We denote by $\mathcal{M}$ the family of bivariate M$^\natural$-concave functions $f : T_\Phi \to \mathbb{R}$ with $f(0,0) = 0$ satisfying the condition (MS$\geq$). We denote by $\mathcal{A}$ the family of bivariate assignment valuations defined on T_Φ. That is,

$$\mathcal{M} = \{f : T_\Phi \to \mathbb{R} \mid f(0,0) = 0,\ f \text{ is an M}^\natural\text{-concave function satisfying (MS}\geq)\},$$
$$\mathcal{A} = \{f : T_\Phi \to \mathbb{R} \mid f \text{ is an assignment valuation in (2.1)}\}.$$

We show that every bivariate assignment valuation satisfies the condition (MS$\geq$), i.e., $\mathcal{A} \subseteq \mathcal{M}$ holds.

Theorem 3.1 (necessity condition)**.** *Let $f : T_\Phi \to \mathbb{R}$ be an assignment valuation in* (2.1)*, and assume that weight function w satisfies the condition* (2.2)*. For every vector $p \in \mathbb{R}^2$, the maximizer set $D_f(p)$ is an M$^\natural$-convex set of positive-type or zero-type.*

Theorem 3.1 follows immediately from the following properties of assignment valuations. It should be noted that the condition (2.2) for the weight function w implies that if a vector $p \in \mathbb{R}^2$ satisfies $p = (w(i,1), w(i,2))$ for some $i \in V$, then such i is uniquely determined.

Lemma 3.2. *Let $f : T_\Phi \to \mathbb{R}$ be an assignment valuation in* (2.1)*, and assume that weight function w satisfies the condition* (2.2)*. Also, let $p \in \mathbb{R}^2$ be a vector such that the maximizer set $D_f(p)$ is two-dimensional.*
(i) *If $p \neq (w(i,1), w(i,2))$ for all $i \in V$, then $D_f(p)$ is of zero-type.*
(ii) *If $p = (w(i,1), w(i,2))$ holds for some $i \in V$, then $D_f(p)$ is of positive-type and satisfies $\ell_{\mathrm{LH}}(D_f(p)) - \ell_{\mathrm{UH}}(D_f(p)) = \varphi(i_p)$ with the (unique) vertex $i_p \in V$ such that $p = (w(i_p,1), w(i_p,2))$.*

Proof of Lemma 3.2 is given in Section 4.

We then show that the inclusion $\mathcal{A} \subseteq \mathcal{M}$ holds with equality. For $f \in \mathcal{M}$, denote by $\mathcal{D}^+(f)$ the family of positive-type maximizer sets of f, and by $P^+(f)$ the set of slope vectors for maximizer sets in $\mathcal{D}^+(f)$. The definitions imply the equation $\mathcal{D}^+(f) = \{D_f(p) \mid p \in P^+(f)\}$.

Theorem 3.3 (sufficiency condition)**.** *Let $f : T_\Phi \to \mathbb{R}_+$ be an M$^\natural$-concave function with $f(0,0) = 0$ satisfying the condition* (MS$\geq$)*. Then, f is an assignment valuation. Moreover, if $P^+(f)$ is given as $\{p_i \mid i \in V\}$ with an appropriately chosen index set V, then f is represented by the complete bipartite graph with vertex set $V \cup \{1,2\}$, weight function $w : V \times \{1,2\} \to \mathbb{R}$, and capacity function $\varphi : V \to \mathbb{Z}_{++}$ given by*

$$w(i,j) = p_i(j) \qquad (i \in V,\ j = 1,2), \tag{3.9}$$
$$\varphi(i) = \ell_{\mathrm{LH}}(D_f(p_i)) - \ell_{\mathrm{UH}}(D_f(p_i))\ (>0) \qquad (i \in V). \tag{3.10}$$

Note that the weight function w given by (3.9) satisfies the condition (2.2) since slope vectors in $P^+(f)$ are all different.

Theorem 3.3 is proved by using the following key lemma, stating that every function $f \in \mathcal{M}$ is uniquely determined by the information about positive-type maximizer sets of f.

Lemma 3.4. *For functions $f, g \in \mathcal{M}$, we have $f = g$ if the following conditions hold:*

$$P^+(f) = P^+(g), \qquad \ell_{\mathrm{LH}}(D_f(p)) - \ell_{\mathrm{UH}}(D_f(p)) = \ell_{\mathrm{LH}}(D_g(p)) - \ell_{\mathrm{UH}}(D_g(p))\ (p \in P^+(f)).$$

Proof of Lemma 3.4 is omitted due to the page limitation.

Proof of Theorem 3.3. Given a function $f \in \mathcal{M}$, let us consider the assignment valuation g represented by the weighted complete bipartite graph in the latter statement of Theorem 3.3, i.e., the assignment valuation $g : T_\Phi \to \mathbb{R}$ is obtained from the complete bipartite graph with vertex set $V \cup \{1, 2\}$, weight function $w : V \times \{1, 2\} \to \mathbb{R}$ in (3.9), and capacity function $\varphi : V \to \mathbb{Z}_{++}$ in (3.10). By Lemma 3.2 applied to g and the definition of g and w, we have $P^+(g) = \{p_i \mid i \in V\} = P^+(f)$. It follows from Lemma 3.2 (ii) that

$$\ell_{\mathrm{LH}}(D_g(p_i)) - \ell_{\mathrm{UH}}(D_g(p_i)) = \varphi(i) = \ell_{\mathrm{LH}}(D_f(p_i)) - \ell_{\mathrm{UH}}(D_f(p_i)) \qquad (i \in V).$$

Hence, we have $f = g$ by Lemma 3.4. □

The following characterization of bivariate assignment valuations can be obtained immediately from Theorems 3.1 and 3.3.

Corollary 3.5. *An $M^\natural$-concave function $f : T_\Phi \to \mathbb{R}$ with $f(0, 0) = 0$ is an assignment valuation if and only if it satisfies the condition* (MS$\geq$).

Based on the characterization of assignment valuations (Theorem 3.3, in particular), we propose an algorithm that determines whether a given function $f : T_\Phi \to \mathbb{R}$ is an assignment valuation or not, and if the answer is "yes," computes its representation.

Algorithm for Checking Assignment Valuation

Step 1: [Check M$^\natural$-concavity] Check whether f satisfies $f(0, 0) = 0$ and the conditions (2.3), (2.4), and (2.5). If f satisfies these conditions, then go to Step 2; otherwise, assert that f is not an assignment valuation.

Step 2: [Check assignment valuation] For each 2-d maximizer set D of f, compute the length of edges $\ell_{\mathrm{LH}}(D)$ and $\ell_{\mathrm{UH}}(D)$. If there exists some D with $\ell_{\mathrm{LH}}(D) < \ell_{\mathrm{UH}}(D)$, (i.e., D is of negative-type), then assert that f is not an assignment valuation; otherwise, go to Step 3.

Step 3: [Compute a weighted bipartite graph] Let $\{p_i \mid i \in V\}$ be the set of slope vectors for positive-type maximizer sets of f with an appropriately chosen index set V. Output the complete bipartite graph with vertex sets $V \cup \{1, 2\}$, weight function $w : V \times \{1, 2\} \to \mathbb{R}$ given by (3.9), and capacity function $u : V \to \mathbb{Z}_{++}$ given by (3.10).

We analyze the running time of the algorithm. Checking the conditions (2.3), (2.4), and (2.5) in Step 1 can be done in $\mathrm{O}(\Phi^2)$ time by using the value oracle for f. For a bivariate M$^\natural$-concave function f, all 2-d maximizer sets and their slope vectors can be computed in $\mathrm{O}(\Phi^2)$ time, as explained below. Once we obtain all 2-d maximizer sets, it is not difficult to compute their edge lengths in $\mathrm{O}(\Phi^2)$ time. Hence, Step 2 requires $\mathrm{O}(\Phi^2)$ time. It is easy to see that Step 3 can be done in $\mathrm{O}(V) = \mathrm{O}(\Phi^2)$ time. Therefore, our algorithm runs in $\mathrm{O}(\Phi^2)$ time in total.

Theorem 3.6. *Given a bivariate function $f : T_\Phi \to \mathbb{R}$, we can determine whether f is an assignment valuation or not in $\mathrm{O}(\Phi^2)$ time. Moreover, if f is an assignment valuation, we can compute the complete bipartite graph with vertex sets $V \cup \{1, 2\}$, weight function $w : V \times \{1, 2\} \to \mathbb{R}$, and capacity function $u : V \to \mathbb{Z}_{++}$ representing f in $\mathrm{O}(\Phi^2)$ time.*

We explain how to compute in $\mathrm{O}(\Phi^2)$ time all 2-d maximizer sets and their slope vectors of a bivariate M$^\natural$-concave function f, where we use the fact that every 2-d M$^\natural$-convex set is given as the union of triangles $T_{\mathrm{ur}}(k, h)$ and $T_{\mathrm{ll}}(k, h)$.

Algorithm for Computing All 2-d Maximizer Sets

Step 1: Let $\mathcal{T}$ be the set of triangles contained in T_Φ, i.e.,

$$\mathcal{T} := \{T_{\rm ur}(k,h) \mid k \geq 1,\ h \geq 1,\ k+h \leq \Phi\} \cup \{T_{\rm ll}(k,h) \mid k \geq 0,\ h \geq 0,\ k+h \leq \Phi - 1\}.$$

For each $T \in \mathcal{T}$, set its slope vector $p_T \in \mathbb{R}^2$ by

$$p_T = \begin{cases} (f(k,h) - f(k-1,h), f(k,h) - f(k,h-1)) & \text{if } T = T_{\rm ur}(k,h), \\ (f(k+1,h) - f(k,h), f(k,h+1) - f(k,h)) & \text{if } T = T_{\rm ll}(k,h). \end{cases}$$

Step 2: If there exists no distinct $T, T' \in \mathcal{T}$ with $p_T = p_{T'}$, then stop. Otherwise, go to Step 3.

Step 3: Select any distinct $T, T' \in \mathcal{T}$ with $p_T = p_{T'}$, delete T and T' from $\mathcal{T}$, and insert $T \cup T'$, where we set $p_{T \cup T'} = p_T$. Go to Step 3.

Step 1 can be done in $O(\Phi^2)$ time. Since a bivariate $M^\natural$-concave function can be extended to a polyhedral concave function, any two triangles contained in T_Φ with the same slope vectors are adjacent. By using this fact, Steps 2 and 3 can be also done in $O(\Phi^2)$ time.

4 Proof of Lemma 3.2

We give a proof of Lemma 3.2. Let $f : T_\Phi \to \mathbb{R}$ be an assignment valuation in (2.1), and assume that weight function w satisfies the condition (2.2). Also, let $p \in \mathbb{R}^2$ be a vector such that the maximizer set $D_f(p)$ is two-dimensional.

By the formula (2.1) for f, the value $\max\{f(k,h) - p(1)k - p(2)h \mid (k,h) \in T_\Phi\}$ is equal to the optimal value of the following optimization problem:

$$\begin{array}{lll} \text{(P)} & \text{Maximize} & \displaystyle\sum_{i \in V}(w(i,1) - p(1))y(i,1) + \sum_{i \in V}(w(i,2) - p(2))y(i,2) \\ & \text{subject to} & y(i,1) + y(i,2) \leq \varphi(i)\ (i \in V), \\ & & y(i,j) \in \mathbb{Z}_+\ (i \in V,\ j = 1,2), \end{array}$$

and the set $D_f(p)$ is represented by using optimal solutions of (P) as follows:

$$D_f(p) = \left\{ x \in \mathbb{Z}^2_+ \;\middle|\; x(j) = \sum_{(i,j) \in V} y(i,j)\ (j = 1,2),\ y \text{ is an optimal solution of (P)} \right\}. \tag{4.11}$$

The problem (P) can be decomposed into $|V|$ independent problems (P_i) ($i \in V$) given as

$$\begin{array}{lll} (\text{P}_i) & \text{Maximize} & w^p(i,1)y(i,1) + w^p(i,2)y(i,2) \\ & \text{subject to} & y(i,1) + y(i,2) \leq \varphi(i),\ y(i,j) \in \mathbb{Z}_+\ (j = 1,2) \end{array}$$

with $w^p(i,j) = w(i,j) - p(j)$ $(j = 1,2)$. Denote by $Y_i^* \subseteq \mathbb{Z}^2_+$ the set of optimal solutions of (P_i), which is given as

$$Y_i^* = \left\{ x \in \mathbb{Z}^2_+ \;\middle|\; \begin{array}{l} x(1) + x(2) \leq \varphi(i), \\ x(1) + x(2) = \varphi(i) \quad \text{if } 0 < \max(w^p(i,1), w^p(i,2)), \\ x(1) = 0 \quad \text{if } w^p(i,1) < \max(0, w^p(i,2)), \\ x(2) = 0 \quad \text{if } w^p(i,2) < \max(0, w^p(i,1)) \end{array} \right\}. \tag{4.12}$$

That is, Y_i^* is the set of vectors $x \in \mathbb{Z}^2_+$ satisfying

$$\begin{array}{ll} x(1) = \varphi(i), \quad x(2) = 0 & \text{if } i \in V_{+1} \equiv \{i' \in V \mid w^p(i',1) > \max(0, w^p(i',2))\}, \\ x(1) = 0, \quad x(2) = \varphi(i) & \text{if } i \in V_{+2} \equiv \{i' \in V \mid w^p(i',2) > \max(0, w^p(i',1))\}, \\ x(1) + x(2) = \varphi(i) & \text{if } i \in V_{+=} \equiv \{i' \in V \mid w^p(i',1) = w^p(i',2) > 0\}, \\ x(1) + x(2) \leq \varphi(i) & \text{if } i \in V_{00} \equiv \{i' \in V \mid w^p(i',1) = w^p(i',2) = 0\}, \\ x(1) = 0, \quad x(2) \leq \varphi(i) & \text{if } i \in V_{-0} \equiv \{i' \in V \mid w^p(i',1) < 0,\ w^p(i',2) = 0\}, \\ x(1) \leq \varphi(i), \quad x(2) = 0 & \text{if } i \in V_{0-} \equiv \{i' \in V \mid w^p(i',1) = 0,\ w^p(i',2) < 0\}, \\ x(1) = x(2) = 0 & \text{if } i \in V_{--} \equiv \{i' \in V \mid w^p(i',1) < 0,\ w^p(i',2) < 0\}. \end{array}$$

We see that y is an optimal solutions of (P) if and only if $(y(i,1), y(i,2)) \in Y_i^*$ holds for every $i \in V$. This relation and (4.11) imply that the set $D_f(p)$ is given as the Minkowski sum of Y_i^* as follows:

$$D_f(p) = \left\{ x \in \mathbb{Z}_+^2 \;\middle|\; x(j) = \sum_{(i,j)\in V} y(i,j) \ (j=1,2),\ (y(i,1), y(i,2)) \in Y_i^* \ (i \in V) \right\} = \sum_{i \in V} Y_i^*$$
$$= \{(x(1), x(2)) \in \mathbb{Z}^2 \mid \lambda_1 \le x(1) \le \mu_1,\ \lambda_2 \le x(2) \le \mu_2,\ \lambda_0 \le x(1) + x(2) \le \mu_0\} \tag{4.13}$$

with

$$\begin{aligned} &\lambda_1 = \varphi(V_{+1}), \qquad \mu_1 = \varphi(V_{+1} \cup V_{+=} \cup V_{00} \cup V_{0-}), \\ &\lambda_2 = \varphi(V_{+2}), \qquad \mu_2 = \varphi(V_{+2} \cup V_{+=} \cup V_{00} \cup V_{-0}), \\ &\lambda_0 = \varphi(V_{+1} \cup V_{+2} \cup V_{+=}), \qquad \mu_0 = \varphi(V \setminus V_{--}). \end{aligned}$$

Here, we denote $\varphi(V') = \sum_{i \in V'} \varphi(i)$ for $V' \subseteq V$.

It follows from (2.8) that

$$\begin{aligned} \ell_{\mathrm{LH}}(D_f(p)) - \ell_{\mathrm{UH}}(D_f(p)) &= \mu_1 + \mu_2 - \mu_0 + \lambda_1 + \lambda_2 - \lambda_0 \\ &= \varphi(V_{+1} \cup V_{+=} \cup V_{00} \cup V_{0-}) + \varphi(V_{+2} \cup V_{+=} \cup V_{00} \cup V_{-0}) - \varphi(V \setminus V_{--}) \\ &\quad + \varphi(V_{+1}) + \varphi(V_{+2}) - \varphi(V_{+1} \cup V_{+2} \cup V_{+=}) \\ &= \varphi(V_{+=} \cup V_{00}) - \varphi(V_{+=}) = \varphi(V_{00}). \end{aligned} \tag{4.14}$$

Suppose that $p \neq (w(i,1), w(i,2))$ for all $i \in V$. Then, we have $V_{00} = \emptyset$, which, together with (4.14), implies $\ell_{\mathrm{LH}}(D_f(p)) - \ell_{\mathrm{UH}}(D_f(p)) = 0$, i.e., $D_f(p)$ is of zero-type.

We then suppose that $p = (w(i,1), w(i,2))$ holds for some $i \in V$. Then, such $i = i_p \in V$ is uniquely determined by the condition (2.2) for the weight function w. Hence, we have $V_{00} = \{i_p\}$, which, together with (4.14), implies $\ell_{\mathrm{LH}}(D_f(p)) - \ell_{\mathrm{UH}}(D_f(p)) = \varphi(i_p) > 0$, i.e., $D_f(p)$ is of positive-type. This concludes the proof of Lemma 3.2.

Remark 4.1. As mentioned in Introduction, the demand information of a bidder in the product-mix auction in Bank of England is represented by a set of weighted bid vectors [9]. We observe below that sets of weighted bid vectors and assignment valuations can represent the same sets of bidders' demand information. Moreover, we show that sets of weighted bid vectors and assignment valuations have a natural one-to-one correspondence.

A weighted bid vector is a pair (b, ω) of a bid $b \in \mathbb{R}^2$ and its weight $\omega \in \mathbb{Z}_{++}$. With a weighted bid vector (b, ω), a demand set $D(p; b, \omega)$ is defined as

$$D(p; b, \omega) = \left\{ x \in \mathbb{Z}_+^2 \;\middle|\; \begin{array}{l} x(1) + x(2) \le \omega, \\ x(1) + y(2) = \omega \quad \text{if } 0 < \max(b(1) - p(1), b(2) - p(1)), \\ x(1) = 0 \quad \text{if } b(1) - p(1) < \max(0, b(2) - p(1)), \\ x(2) = 0 \quad \text{if } b(2) - p(2) < \max(0, b(1) - p(1)) \end{array} \right\}. \tag{4.15}$$

Using a set of weighted bid vectors $\mathcal{B} = \{(b_i, \omega(i)) \mid i \in V\}$ with an appropriately chosen index set V, we represent a demand set $D^{\mathcal{B}}(p)$ given as the Minkowski sum of $D(p; b_i, \omega(i))$:

$$D^{\mathcal{B}}(p) = \sum_{i \in V} D(p; b_i, \omega(i)). \tag{4.16}$$

We see from (4.12), (4.13), (4.15), and (4.16) that sets of weighted bid vectors can represent the same sets of bidders' demand information as assignment valuations. Indeed, for a set of weighted bid vectors $\mathcal{B} = \{(b_i, \omega(i)) \mid i \in V\}$, define a complete bipartite graph $G = (V, \{1,2\}; V \times \{1,2\})$ with weight function $w : V \times \{1,2\} \to \mathbb{R}$ and supply function $\varphi : V \to \mathbb{Z}_{++}$ given as

$$w(i,j) = b_i(j) \quad (i \in V,\ j = 1,2), \qquad \varphi(i) = \omega(i) \quad (i \in V),$$

and let $f : T_\Phi \to \mathbb{R}$ be the associated assignment valuation. Then, it follows from (4.12), (4.13), (4.15), and (4.16) that $D^{\mathcal{B}}(p) = D_f(p)$ for every $p \in \mathbb{R}^2$. This shows that sets of weighted bid vectors have a natural one-to-one correspondence with weighted complete bipartite graphs representing assignment valuations. □

References

[1] Baldwin, E., Klemperer, P.: Proof that the product-mix auction bidding language can represent any substitutes preferences. Working Paper 2021–W05. Nuffield College (2021)

[2] Baldwin, E., Bichler, M., Fichtl, M., Klemperer, P.: Strong substitutes: Structural properties, and a new algorithm for competitive equilibrium prices. Mathematical Programming, published online (2022)

[3] Baldwin, E., Goldberg, P.W., Klemperer, P., Lock, E.: Solving strong-substitutes product-mix auctions. arXiv preprint arXiv:1909.07313 (2019)

[4] Fujishige, S., Yang, Z.: A note on Kelso and Crawford's gross substitutes condition. Mathematics of Operations Research 28, 463–469 (2003)

[5] Goldberg, P.W., Lock, E., Marmolejo-Cossío, F.: Learning strong substitutes demand via queries. Proceedings of International Conference on Web and Internet Economics, 401–415 (2020)

[6] Gul, F., Stacchetti, E.: Walrasian equilibrium with gross substitutes. Journal of Economic Theory 87. 95–124 (1999)

[7] Gul, F., Stacchetti, E.: The English auction with differentiated commodities. Journal of Economic Theory 92, 66–95 (2000).

[8] Kelso, A.S., Crawford, V.P.: Job matching, coalition formation and gross substitutes. Econometrica 50, 1483–1504 (1982)

[9] Klemperer, P.: The product-mix auction: A new auction design for differentiated goods. Journal of the European Economic Association 8, 526–536 (2010)

[10] Milgrom, P., Strulovici, B.: Substitute goods, auctions, and equilibrium. Journal of Economic Theory 144, 212–247 (2009)

[11] Murota, K.: Convexity and Steinitz's exchange property. Advances in Mathematics 124, 272–311 (1996)

[12] Murota, K.: Discrete convex analysis. Mathematical Programming 83, 313–371 (1998)

[13] Murota, K.: Discrete Convex Analysis. SIAM, Philadelphia (2003)

[14] Murota, K.: Discrete convex analysis: A tool for economics and game theory. Journal of Mechanism and Institution Design 1, 151–273 (2016)

[15] Murota, K., Shioura, A.: M-convex function on generalized polymatroid. Mathematics of Operations Research 24, 95–105 (1999)

[16] Shapley, L.: Complements and substitutes in the optimal assignment problem. Naval Research Logistics Quarterly 9, 45–48 (1962)

[17] Shioura, A., Tamura, A.: Gross substitutes condition and discrete concavity for multi-unit valuations: A survey. Journal of the Operations Research Society of Japan 58, 61–103 (2015)

On vertex-coloring {a,b}-edge-weightings of graphs

Péter Madarasi

Department of Operations Research, ELTE Eötvös Loránd University, and the ELKH-ELTE Egerváry Research Group on Combinatorial Optimization, Eötvös Loránd Research Network (ELKH), Pázmány Péter sétány 1/C, 1117 Budapest, Hungary.
madarasip@staff.elte.hu

Máté Simon

Department of Operations Research ELTE Eötvös Loránd University, and the MTA-ELTE Momentum Matroid Optimization Research Group, Pázmány Péter sétány 1/C, 1117 Budapest, Hungary
matesimon@student.elte.hu

Abstract: For a given graph $G = (V, E)$, an $\{a, b\}$-edge-weighting is an assignment $w : E \to \{a, b\}$, which is called proper if the induced labeling $z : V \to \mathbb{Z}$ is a proper vertex coloring of G, where a and b are distinct integers, and $z(v) = \sum_{e \in \Delta(v)} w(e)$.

Thomassen, Wu and Zhang gave a polynomial-time algorithm for deciding whether a given bipartite graph has a proper $\{1, 2\}$-edge-weighting. We consider the natural generalization of this problem when a partial edge-weighting is to be extended, which is proven to be NP-complete for any distinct integers a and b. For trees, however, the problem is shown to be solvable in polynomial time, which implies an alternative polynomial-time algorithm for the so-called antifactor problem and also for deciding whether a tree has a $\{0, 1\}$-edge-weighting.

Dudek and Wajc proved that deciding whether a given graph G has a proper $\{1, 2\}$-edge-weighting is NP-complete. Strengthening their result, we show that the problem is NP-complete for any distinct integers a and b.

Keywords: 1-2-3 conjecture, $\{a, b\}$-edge-weighting, NP-completeness, Irregular graphs, Graph coloring

1 Introduction

Throughout this paper, $G = (V, E)$ denotes a simple, finite, undirected graph. A $\{1, \ldots, k\}$-*edge-weighting* is an assignment w which assigns numbers from the set $\{1, \ldots, k\}$ to the edges of G. We say that an edge-weighting is *proper* or *feasible* if the induced vertex coloring $z : V \to \mathbb{Z}$, where $z(v) = \sum_{e \in \Delta(v)} w(e)$, is a proper coloring, that is $z(u) \neq z(v)$ holds for every edge $uv \in E$. If G has a proper $\{1, 2, 3\}$-edge-weighting, then we say that G has the 1-2-3 *property*, which can be defined similarly for any other weight set as well.

Karoński, Łuczak and Thomason formulated the so-called 1-2-3 conjecture in 2004 [15], which states that every simple graph without isolated edges has the 1-2-3 property. This conjecture fostered several new interesting questions. The focus of the present paper is on one of these questions, the existence of $\{a, b\}$-edge-weightings. For a more detailed treatment of these results, the reader is referred to the full version of the paper [19].

In 2016, Thomassen, Wu and Zhang [23] proved that a bipartite graph has the 1-2 property if and only if it is not a so-called odd multi-cactus, which implies that one can decide in polynomial time whether

[1]The work was supported by the Lendület Programme of the Hungarian Academy of Sciences – grant number LP2021-1/2021, and the Ministry of Innovation and Technology of Hungary from the National Research, Development and Innovation Fund, financed under the ELTE TKP 2021-NKTA-62 funding scheme. The research was supported by the Ministry of Innovation and Technology NRDI Office within the framework of the Artificial Intelligence National Laboratory Program.

a bipartite graph has the 1-2 property. In fact, their approach also extends to $\{a, b\}$-edge-weightings provided that $a < b$, a is odd and b is even. Recently, Lyngise showed that exactly the odd multi-cacti have no proper edge-weightings for 2-connected bipartite graphs when a is odd and $b = a+2$ [5], and also for bridgeless bipartite graphs when $a = 0$ and $b = 1$ [18]. Based on these positive results, we investigate whether a partial edge-weighting can be extended to a proper edge-weighting. In Section 2, we show that this more general problem is NP-complete even for bipartite graphs, however, it is polynomial-time solvable for trees. The latter statement will be proven by giving an efficient dynamic programming algorithm. As a special case, this implies an alternative polynomial-time algorithm for the so-called antifactor problem [16] and also for deciding whether a tree has the 0-1 property, which was first solved in [18].

Furthermore, in 2011, Dudek and Wajc [11] proved that deciding whether a given graph has the 1-2 property is NP-complete. To the best of our knowledge, the more general problem, when we ask if a feasible $\{a, b\}$-edge-weighting exists, remained open thus far. Section 3 investigates an extension of the results of Dudek and Wajc, we show that their statement also holds for arbitrary a and b.

The next section gives a brief overview of some further problems and results related to the 1-2-3 conjecture.

1.1 Motivation and previous results

The question of the existence of $\{a, b\}$-edge-weightings was inspired by the 1-2-3 conjecture, which itself comes from the study of graph irregularity. By simple graph-theoretic observations, one can easily show that there exists no "opposite" of a simple regular graph, that is, a simple graph with all-different degrees. Chartrand et al. [6] investigated the smallest value k such that by replacing each edge with at most k parallel edges, the resulting multigraph G' becomes irregular (that is each node has a different degree). The minimum value of k is called the irregularity strength of G. For further information on this topic see [12], [4] and [20]. Another possible approach is when we do not require that all nodes in the resulting multigraph have different degrees, but only that the degrees of the adjacent nodes are different. Notice that instead of edge multiplication, we can look for a proper $\{1, \ldots, k\}$-edge-weighting. Exchanging the weight set $\{1, \ldots, k\}$ to $\{a, b\}$, we obtain the $\{a, b\}$-edge-weighting problem.

Early articles and results, such as in which the 1-2-3 conjecture was first introduced [15], focus on the relationship between $\chi(G)$ and $\chi_{\sum}(G)$, where $\chi_{\sum}(G)$ is the smallest integer k for which a proper $\{1, \ldots, k\}$-edge-weighting exists in G. One of the first results from [15] states that if $(\Gamma, +)$ is a finite abelian group of odd order, and G is a $|\Gamma|$-colorable graph without isolated edges, then there exists an edge-weighting of G with the elements of Γ such that the induced vertex coloring is proper.

Further results in connection with the chromatic number: If G is 2-connected and $\chi(G) \geq 3$, then $\chi_{\sum}(G) \leq \chi(G)$ [22]. Moreover, for every integer $k \geq 3$ and any graph G without isolated edges, the following hold: 1) If G is k-colorable for odd k, then $\chi_{\sum}(G) \leq k$ [15]. 2) If G is k-colorable for $k \equiv 0 \pmod 4$, then $\chi_{\sum}(G) \leq k$ [10]. 3) If G is k-colorable, 2-connected and has minimum degree at least $k+1$ for $k \equiv 2 \pmod 4$, then $\chi_{\sum}(G) \leq k$ [17].

The first general upper bound for $\chi_{\sum}(G)$ was given by Addario-Berry, Dalal, McDiarmid, Reed and Thomason [1], who proved that $\chi_{\sum}(G) \leq 30$. Their method is based on the investigation of the so-called degree-constrained subgraph problem, which was further refined by Addario-Berry, Dalal and Reed [2], who managed to improve this upper bound to 16, then Wang and Yu [24] further improved it to 13. The best known upper bound is due to Kalkowski, Karoński and Pfender [14], who proved that $\chi_{\sum}(G) \leq 5$ holds. In other words, every graph without isolated edges has the 1-2-3-4-5 property.

Moreover, it is also known that the 1-2-3 conjecture holds if G is large and dense enough: there exists a constant n' such that every graph $G = (V, E)$ with at least n' nodes has the 1-2-3 property if the degree of every node is at least $0.099985|V|$ [25]. Furthermore, it is known that if G is a random graph (according to the Erdős-Rényi model), then it has the 1-2 property asymptotically almost surely, see [2]. If we restrict ourselves to regular graphs, then Jakob Przybylo achieved the most significant progress [21], namely, every regular graph has the 1-2-3-4 property, and the 1-2-3 conjecture holds if $d \geq 10^8$ and G is

d-regular. On the other hand, Dudek and Wajc [11] showed that deciding whether a given graph has the 1-2 property is NP-complete. For bipartite graphs, however, this problem can be solved in polynomial time based on the result of Thomassen, Wu and Zhang [23].

One might also define other kinds of weightings. For example, in the node-weighting problem, we want to assign weights to the nodes (instead of the edges) and the labels of the nodes are defined as the sum of the weights of their neighbors. It was shown in [3] that deciding whether a graph G has a proper node-weighting from the set $\{1, \ldots, k\}$ is NP-complete for any $k \geq 2$. This result holds even if we restrict ourselves to 3-colorable planar graphs and $k = 2$. Furthermore, it is also NP-complete for 3-regular graphs in case of $k = 2$ [9].

Similar problems can be obtained by modifying the definition of the labels of the nodes. For example, one can take the product of the weights instead of their sum. This way, we obtain the problems called *edge-weighting by product* and *node-weighting by product*. Let us briefly summarize some of the hardness results related to these problems. It is NP-complete to decide whether a given 3-regular planar graph has a proper edge-weighting by product from the set $\{1, 2\}$ [8]. It was shown in [8] that deciding the existence of $\{1, 2\}$-node-weighting by product is NP-complete for 3-colorable planar graphs. Moreover, if we omit the planarity and colorability conditions but the weights can be chosen from the set $\{1, \ldots, k\}$ for some $k \geq 3$, then we still get an NP-complete problem.

2 Extending partial edge-weightings

Thomassen, Wu and Zhang [23] proved in 2016 that deciding whether a given bipartite graph has the 1-2 property is possible in polynomial time, while the same problem for arbitrary graphs is NP-complete [11]. Motivated by the former statement, this section investigates whether a partial $\{a, b\}$-edge-weighting can be extended on bipartite graphs, where by a partial $\{a, b\}$-edge-weighting we mean that on a subset of the edges we fix the labels in advance. First of all, let us outline the basic problem, which has not been addressed in the literature yet, as far as we know.

Problem 1 *Given a graph G with some of its edges already weighted from set $\{a, b\}$, where a and b are two distinct rational numbers. The question is if we can assign weights from $\{a, b\}$ to the uninitialized edges such that the induced coloring is proper.*

Theorem 2 *Problem 1 is NP-complete for bipartite graphs.*

This theorem can be proven by a reduction from the NP-complete degree-prescribed subgraph problem. In this problem, the goal is to find a subgraph $H = (V, E')$ of a given graph $G = (V, E)$ such that the degree of every node v in H is from a predefined degree set $F_v \subseteq \{0, \ldots, d_G(v)\}$, that is, $d_H(v) \in F_v$ for every v in V. For a given graph G and degree prescription F, we create a graph G' along with a partial edge-weighting such that the remaining edges can be weighted properly if and only if the given instance is solvable.

2.1 Extendability on trees

Theorem 2 shows that Problem 1 is NP-complete on bipartite graphs. In this section, we investigate the same problem on trees, and we give a polynomial-time algorithm which, for a given tree and integers a and b, either completes a given partial $\{a, b\}$-edge-weighting or concludes that no such weighting exists. As a special case, one obtains a new method to decide whether a tree has the 0-1 property, which was first shown to be polynomial-time solvable in [18].

Theorem 3 *Problem 1 can be solved in polynomial time on trees for any distinct integers a and b.*

Proof: We give a dynamic programming algorithm which either extends the partial $\{a, b\}$-edge-weighting into a feasible one or concludes that it cannot be extended. Let us appoint one of the leaf nodes as the

root of the tree and let T_v denote the subtree beneath v for every node $v \in V$. For every edge uv, let $L_{uv} \subseteq \{a, b\}$ denote the set of the allowed weights for uv based on the partially initialized edge-weighting. We want to decide whether T_v can be extended feasibly such that we fix the weight of uv and the sum of the weights on the edges incident to v, where u is closer to the root than v is. Formally, for every edge uv, we define a subproblem $f(uv)$ as the set of those pairs $(k, l) \in \mathbb{Z} \times \{a, b\}$ for which there exists a weighting of T_v such that $w(uv) = l \in L_{uv}$ and $z(v) = k - l$.

For a given edge uv, let e_i denote the edge between v and its children v'_i for $i = 1, \ldots, d(v) - 1$. Notice that $(k, l) \in f(uv)$ if and only if the following two conditions hold:

1. For every $i = 1, \ldots, d(v) - 1$, there exists a weight $l_i \in L_{e_i}$ and label $k_i \in \mathbb{Z} \backslash \{k\}$ such that $(k_i, l_i) \in f(e_i)$, and
2. $\sum l_i = k - l$,

which gives a way to recursively compute $f(uv)$ in polynomial time, because these conditions can be checked efficiently as follows. There exist unique integers α and β for which

$$\begin{aligned} a \cdot \alpha + b \cdot \beta &= k - l \\ \alpha + \beta &= d(v) - 1 \end{aligned} \tag{1}$$

hold, because $(k, l) \in f(uv)$. That is, exactly α out of $e_1, \ldots, e_{d(v)-1}$ have weight a, and β are weighted b. Let $L^k_{e_i} \subseteq L_{e_i}$ denote the possible weights of e_i if $z(v'_i) \neq k$, and observe that $(k, l) \in f(uv)$ if and only if $L^k_{e_i} \neq \emptyset$, $|\{i : a \in L^k_{e_i}\}| \geq \alpha$ and $|\{i : b \in L^k_{e_i}\}| \geq \beta$ hold for every $i = 1, \ldots, d(v) - 1$. For any e_i and k, $L^k_{e_i}$ can be computed easily by iterating through $f(e_i)$, therefore we obtain an algorithm for computing $f(uv)$ running in $O(n^2)$ steps, provided that the subproblems are computed in increasing order by the depth of the subtrees T_v.

For the base case of the recursion, if subtree T_v consists of a single node for uv, then $f(uv) = \{(l, l) : l \in L_{uv}\}$ by definition.

Once $f(uv)$ is computed for all $uv \in E$, there exists a feasible extension of the partial edge-weighting if and only if there exists $(k, l) \in f(e)$ such that $k \neq l$, where e is the leaf edge incident to the root. This means that the labels of the two endpoints of e are different in the weighting provided by the fact that $(k, l) \in f(e)$. Otherwise, if $k = l$ for all $(k, l) \in f(e)$, then $f(e)$ is either empty or the endpoints of e have the same label in each feasible weighting, which means that no feasible extension of the partial edge-weighting exists. Computing a subproblem $f(uv)$ takes $O(n^2)$ steps, hence the total running time of the algorithm is $O(n^3)$. □

Note that Theorem 3 easily extends to the minimum-cost version of the problem in which each weight-edge assignment has an associated cost, and the total cost of the $\{a, b\}$-edge-weighting is to be minimized.

2.1.1 A special case — the antifactor problem

This section investigates a variant of the degree prescribed subgraph problem, the so-called antifactor problem, where exactly one degree is prohibited at every node. That is, given a connected graph G and an assignment $f : V \to \mathbb{Z}_+$, we look for a subgraph H of G, such that $d_H(v) \neq f(v)$ for all $v \in V$. This problem was solved by Lovász [16] and it was further generalized in [7] and [13]. It is easy to show that the problem is always solvable when the graph has a cycle, therefore, we can assume that G is a tree.

We prove that the antifactor problem for trees can be solved by the dynamic programming algorithm given in the proof of Theorem 3, meaning that we get an alternative polynomial-time algorithm for the antifactor problem.

Theorem 4 *The antifactor problem for trees can be reduced to the problem of deciding whether a given tree has a proper* $\{0, 1\}$*-edge-weighting.*

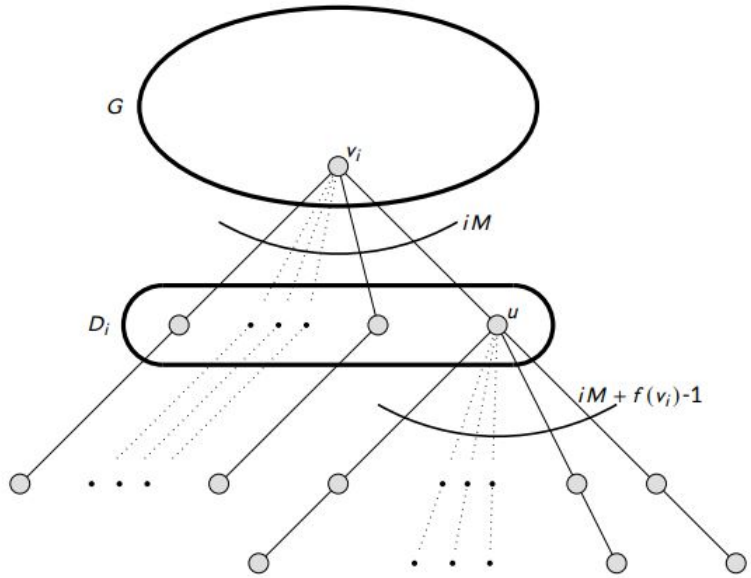


Figure 1: Illustration of the construction described in the proof of Theorem 4.

PROOF: Let us take an instance of the antifactor problem, that is, we are given a tree G and an assignment $f : V \to \mathbb{Z}_+$. We construct a tree G' in polynomial time which has a proper $\{0, 1\}$-edge-weighting if and only if the given instance of the antifactor problem is solvable.

We start by taking a copy of G, then, for every $v_i \in V$, we modify G' as follows. Add iM new leaf nodes connected only to v_i, where $M = n + 1$. Let D_i denote the set of these newly added vertices. Let u denote one of the nodes in D_i, and add $(iM + f(v_i) - 1)$ new path of length 2 connected to u, and add a leaf edge to each node in D_i other than u. Figure 1 illustrates the construction for v_i. Clearly, one can construct G' in polynomial time. In the rest of this proof, we show that the antifactor problem for G and f can be solved if and only if G' has a proper $\{0, 1\}$-edge-weighting.

First, assume that the antifactor problem for G and f is solvable, and let H denote a feasible solution. To create a feasible $\{0, 1\}$-edge-weighting of G', set the labels of the edges of G' in the following fashion: for each edge $e \in E_G$, if H contains e, then let $w(e)$ be 1, otherwise 0. Weight all the remaining edges — which are exactly the newly added edges — with 1. We show that this $\{0, 1\}$-edge-weighting is proper. In G', there cannot be a collision between two vertices of the original graph, because the induced color of any two nodes of the original graph are different: Since the contribution of the newly added edges is exactly iM and that of the edges in the original graph is at most $(n - 1)$, the largest possible label of v_i is $U_i = iM + n - 1 = (i + 1)n + i - 1$. The smallest possible label of v_{i+1}, on the other hand, is $L_{i+1} = (i+1)M = (i+1)n+i+1$, which can be shown using a similar argument. We can see that L_{i+1} is strictly greater than U_i, that is, $z_{v_{i+1}} > z_{v_i}$ in any proper $\{0, 1\}$-edge-weighting for all $i \in \{1, \ldots, n - 1\}$, since $L_{i+1} \leq z_{v_{i+1}}$ and $U_i \geq z_{v_i}$.

Observe that there is no collision between v_i and the nodes in D_i either, because H was a feasible solution to the antifactor problem, thus the number of edges incident to v_i with weight 1 inside the original graph G cannot be equal to $f(v_i)$ by the setting of the weights. One can easily see that collisions along the rest of the newly added edges are impossible as well.

Second, assume that G' has a proper $\{0, 1\}$-edge-weighting. Then let H be the subgraph of G consisting of the edges with weight 1. We prove that H is a feasible solution to the antifactor problem for G and f. Each edge e between v_i and $D_i \setminus \{u\}$ must be weighted with 1, because each node in $D_i \setminus \{u\}$ has an incident leaf edge, along which a collision would occur if the weight of e would be 0. Similar argument shows that all edges in $\Delta(u) \setminus \{v_iu\}$ must have weight 1. Therefore, $z(v_i) = d_H(v_i)+|D_i \setminus \{u\}|+w(v_iu) = d_H(v_i)+iM-1+w(v_iu)$ and $z(u) = |\Delta(u)\setminus\{v_iu\}|+w(v_iu) = iM+f(v_i)-1+w(v_iu)$, and by $z(v_i) \neq z(u)$,

we have that $d_H(v_i) \neq f(v_i)$, which completes the proof. □

The tree obtained by the construction given in the proof of Theorem 4 has a $\{0,1\}$-edge-weighting if and only if the corresponding antifactor problem has a feasible solution. As Lovász gave an elegant characterisation for the latter, we also have a characterisation of the trees obtained by the construction with the 0-1 property. We leave it open whether this generalizes to a simple characterisation for trees in general.

3 $\{a,b\}$-edge-weightings in general graphs

In 2011, Dudek and Wajc [11] proved that deciding whether a given graph G has the 1-2 property is NP-complete. In this section, we consider a generalization of this problem, when we change the weight set from $\{1,2\}$ to $\{a,b\}$ for arbitrary distinct a and b.

The following claim shows that we can restrict ourselves to the case when a and b are integers and relative primes.

Claim 5 *Let a,b be a rational pair. Then for every $d \neq 0$, there is one-to-one correspondence between proper $\{a,b\}$-edge-weightings and proper $\{ad,bd\}$-edge-weightings.*

This simple claim holds, since multiplication by $d \neq 0$ on all edges does not change the feasibility of an edge-weighting. We say that a and b are *relevant* if they are integers, relative primes, at most one of them is negative, $a \neq b$ and $|b| \geq |a|$. By Claim 5, we can assume without loss of generality that a,b are relevant whenever we consider $\{a,b\}$-edge-weightings.

One of the main theorems we have proven in [19] is the following:

Theorem 6 *Let a and b be relevant numbers, and let G be an arbitrary simple graph. Then, it is NP-complete to decide whether a proper $\{a,b\}$-edge-weighting exists.*

The proof of this theorem consists of two parts, which we state in the next two theorems.

Theorem 7 *Let a and b be relevant numbers such that $a \neq -1$ and $b \neq 1$ holds. Then, it is NP-complete to decide whether a proper $\{a,b\}$-edge-weighting exists for simple graphs.*

This theorem can be proven by a reduction from the NP-complete 3-COLOR problem, similarly to the original proof of Dudek and Wajc. The key idea is to construct a so called a-forcing gadget, which is a graph that has a leaf edge which must be weighted with a in every proper $\{a,b\}$-edge-weighting. Using this gadget, for a graph G, one can construct another graph G' such that the induced label of any node in any proper $\{a,b\}$-weighting is one of three possible labels — defining a proper vertex coloring on the original graph G and vice-versa. The other step in the proof of Theorem 6 is as follows.

Theorem 8 *Let $G = (V,E)$ be a simple graph. It is NP-complete to decide whether G has a $\{-1,1\}$-edge-weighting.*

For this theorem, a fundamentally different reduction can be given from the NP-complete NAE3-SAT3 problem, since some of the gadgets given in the proof of Theorem 7 were meaningless in this case.

4 Concluding remarks

This paper presented some progress in terms of the hardness of finding $\{a,b\}$-edge-weightings, and proposed the question of the extendability of a partial edge-weighting in bipartite graphs.

Generalizing the result of Dudek and Wajc [11], we proved that it is NP-complete to decide whether a graph has a proper $\{a,b\}$-edge-weighting. As a generalization of the $\{a,b\}$-edge-weighting problem on bipartite graphs, we asked whether a partial $\{a,b\}$-edge-weighting of a bipartite graph can be extended. This problem was shown to be NP-complete, and a polynomial-time algorithm was given for trees. As a

special case, this implies an alternative polynomial-time algorithm for the so-called antifactor problem [16] and also for deciding whether a tree has the 0-1 property [18].

Another interesting question is whether the dynamic programming algorithm given in Section 2.1 extends to graphs of bounded tree-width or for other types of labelings, for example when the label of a node is defined as the product of the weights written on the incident edges.

It remains open whether the connection of the $\{0,1\}$-property and the antifactor problem generalizes to a nice characterisation for the trees with the $\{0,1\}$-property.

When $a < b$, a is odd and b is even, Thomassen, Wu and Zhang [23] proved that a bipartite graph has the a-b property if and only if it is an odd multi-cactus. Lyngise showed that the same holds for 2-connected bipartite graphs when a is odd and $b = a + 2$ [5], and also for bridgeless bipartite graphs when $a = 0$ and $b = 1$ [18]. The general case, however, remains open. Based on computer testing all bipartite graphs with at most 14 nodes, we conjecture that exactly the odd multi-cacti have no proper $\{a, b\}$-edge-weightings whenever $0 < a < b$.

References

[1] L. Addario-Berry, K. Dalal, C. McDiarmid, B. A. Reed, and A. Thomason. Vertex-colouring edge-weightings. *Combinatorica*, 27(1):1–12, 2007.

[2] L. Addario-Berry, K. Dalal, and B. A. Reed. Degree constrained subgraphs. *Electron. Notes Discret. Math.*, 19:257–263, 2005.

[3] A. Ahadi, A. Dehghan, M.-R. Kazemi, and E. Mollaahmadi. Computation of lucky number of planar graphs is NP-hard. *Information processing letters*, 112(4):109–112, 2012.

[4] M. Aigner and E. Triesch. Irregular assignments of trees and forests. *SIAM Journal on Discrete Mathematics*, 3(4):439–449, 1990.

[5] J. Bensmail, F. Mc Inerney, and K. S. Lyngsie. On {a, b}-edge-weightings of bipartite graphs with odd a, b. *Discussiones Mathematicae Graph Theory*, 42(1):159–185, 2022.

[6] G. Chartrand, M. S. Jacobson, J. Lehel, O. R. Oellermann, S. Ruiz, and F. Saba. Irregular networks. *Congr. Numer*, 64(197-210):250th, 1988.

[7] G. Cornuéjols. General factors of graphs. *J. Comb. Theory Ser. B*, 45(2):185–198, aug 1988.

[8] A. Dehghan, M.-R. Sadeghi, and A. Ahadi. Algorithmic complexity of proper labeling problems. *Theoretical Computer Science*, 495:25–36, 2013.

[9] A. Dehghan, M.-R. Sadeghi, and A. Ahadi. The complexity of the sigma chromatic number of cubic graphs. *arXiv preprint arXiv:1403.6288*, 2014.

[10] Y. Duan, H. Lu, and Q. Yu. L-factors and adjacent vertex-distinguishing edge-weighting. *East Asian Journal on Applied Mathematics*, 2(2):83–93, 2012.

[11] A. Dudek and D. Wajc. On the complexity of vertex-coloring edge-weightings. *Discrete Mathematics and Theoretical Computer Science*, 13(3):45–50, 2011.

[12] R. Faudree and J. Lehel. Bound on the irregularity strength of regular graphs. In *Colloq Math Soc Janos Bolyai*, volume 52, pages 247–256, 1987.

[13] A. Frank, L. Chi Lau, and J. Szabó. A note on degree-constrained subgraphs. *Discrete Mathematics*, 308(12):2647–2648, 2008.

[14] M. Kalkowski, M. Karoński, and F. Pfender. Vertex-coloring edge-weightings: towards the 1-2-3-conjecture. *Journal of Combinatorial Theory, Series B*, 100(3):347–349, 2010.

[15] M. Karoński, T. Łuczak, and A. Thomason. Edge weights and vertex colours. *Journal of Combinatorial Theory Series B*, 91(1):151–157, 2004.

[16] L. Lovász. Antifactors of graphs. *Periodica Mathematica Hungarica*, 4(2-3):121–123, 1973.

[17] H. Lu, X. Yang, and Q. Yu. On vertex-coloring edge-weighting of graphs. *Frontiers of Mathematics in China*, 4(2):325–334, 2009.

[18] K. S. Lyngsie. On neighbour sum-distinguishing $\{0, 1\}$-edge-weightings of bipartite graphs. *Discrete Mathematics & Theoretical Computer Science*, 20, 2018.

[19] P. Madarasi and M. Simon. On vertex-coloring $\{a, b\}$-edge-weightings of graphs. Technical Report TR-2022-06, Egerváry Research Group, Budapest, 2022. `egres.elte.hu`.

[20] T. Nierhoff. A tight bound on the irregularity strength of graphs. *SIAM Journal on Discrete Mathematics*, 13(3):313–323, 2000.

[21] J. Przybyło. The 1–2–3 conjecture almost holds for regular graphs. *Journal of Combinatorial Theory, Series B*, 147:183–200, 2021.

[22] B. Seamone. The 1-2-3 conjecture and related problems: a survey. *arXiv preprint arXiv:1211.5122*, 2012.

[23] C. Thomassen, Y. Wu, and C.-Q. Zhang. The 3-flow conjecture, factors modulo k, and the 1-2-3-conjecture. *Journal of Combinatorial Theory, Series B*, 121:308–325, 2016.

[24] T. Wang and Q. Yu. On vertex-coloring 13-edge-weighting. *Frontiers of Mathematics in China*, 3(4):581–587, 2008.

[25] L. Zhong. The 1-2-3-conjecture holds for dense graphs. *Journal of Graph Theory*, 90(4):561–564, 2019.

On the generalized Mycielskian of complements of odd cycles

Anna Gujgiczer[1]

Department of Computer Science and Information Theory
Budapest University of Technology and Economics
and
MTA-BME Lendület Arithmetic Combinatorics Research Group, ELKH, Budapest, Hungary
gujgicza@cs.bme.hu

Gábor Simonyi[2]

Alfréd Rényi Institute of Mathematics, Budapest, Hungary
and
Department of Computer Science and Information Theory
Budapest University of Technology and Economics
simonyi@renyi.hu

Gábor Tardos

Alfréd Rényi Institute of Mathematics, Budapest, Hungary
tardos@renyi.hu

Abstract: The main goal of this talk is to popularize a (special case of a) result of Pan and Zhu according to which whether the generalized Mycielski construction applied to the complement $\overline{C_{2k+1}}$ of an odd cycle makes the chromatic number increase or not depends on the residue of $2k+1$ modulo 4. This surprizing phenomenon is explained by the topological properties of the circular complete graphs $K_{p/q}$ and the trivial observation that $\overline{C_{2k+1}}$ is isomorphic to $K_{(2k+1)/2}$.

Keywords: Mycielsky construction, circular coloring, topological method

1 Introduction

The Mycielski construction is one of the best-known constructions that from any graph G creates a graph $M(G)$ with the same clique number and larger chromatic number. Formally, denoting the chromatic number of a graph F by $\chi(F)$ and its clique number by $\omega(F)$, we have

$$\chi(M(G)) = \chi(G) + 1, \quad \text{while} \quad \omega(M(G)) = \omega(G).$$

The generalized Mycielski construction has a further integer parameter r (describing the number of "levels" in the construction) and it creates the graph $M_r(G)$ from a given graph G as follows.

Definition 1 *For a graph G and positive integer r the generalized Mycielskian $M_r(G)$ of G is defined by*

$$V(M_r(G)) = \{(i, v) \colon 0 \le i \le r-1, v \in V(G)\} \cup \{z\};$$

[1]Research is partially supported by the National Research, Development and Innovation Office (NKFIH) grant K–120706 of NKFIH Hungary.

[2]Research is partially supported by the National Research, Development and Innovation Office (NKFIH) grants K–120706, K–132696 and SNN-135643 of NKFIH Hungary.

[3]Research is partially supported by the National Research, Development and Innovation Office (NKFIH) grants K–132696 and SNN-135643 of NKFIH Hungary.

$E(M_r(G)) = \{\{(i,u),(j,v)\}\colon \{u,v\} \in E(G) \text{ and } (i=j=0 \text{ or } |i-j|=1)\} \cup \{\{z,(r-1,v)\}\colon v \in V(G)\}$.

The (usual) Mycielskian of graph G is the special case of $r = 2$, i.e., $M(G) = M_2(G)$.

It is easy to see that we always have $\chi(M_r(G)) \leq \chi(G) + 1$, but unlike in the cases $r \leq 2$ when this inequality always holds with equality, for $r \geq 3$ we can have $\chi(M_r(G)) = \chi(G)$ as well. For example, for the complementary graph $\overline{C_7}$ of the 7-cycle we have

$$\chi(M_3(\overline{C_7})) = 4 = \chi(\overline{C_7}).$$

This example appears, for example, in Tardif's paper [11] and this is even used in an essential way in his more recent paper [12], where the author gave new counterexamples to the famous Hedetniemi conjecture for much smaller chromatic numbers than it was done before and he used the above phenomenon to improve his construction a little further. (Since then this record was yet further improved first by Wrochna [14] and then by Tardif [13] where he achieved the best possible such value. The latter ones did not use the generalized Mycielski construction.)

On the other hand, it was proved by Stiebitz [10] that if we apply the generalized Mycielski construction iteratively starting with an odd cycle (in fact, one can also start with a single edge and then obtain an odd cycle after the first iteration), then the chromatic number increases at every step. Since [10] is not easily available, this result is given with proof also in [1], cf. also [5].

Recently we observed that for the complementary graph $\overline{C_{2k+1}}$ of an odd cycle of length at least 5 and large enough r whether we have

$$\chi(M_r(\overline{C_{2k+1}})) = k + 1 = \chi(\overline{C_{2k+1}})$$

or

$$\chi(M_r(\overline{C_{2k+1}})) = k + 2 = \chi(\overline{C_{2k+1}}) + 1$$

seems to depend on the residue of the length of the complementary odd cycle modulo 4. With some effort we managed to verify that this is indeed the case but later realized that we have just rediscovered a special case of a more general result due to Pan and Zhu [7]. This made us feel that this result is not known well enough, and especially not in the (by our opinion) both very appealing and surprising form of this special case. This is why we would like to popularize it.

2 Complements of odd cycles, circular chromatic number and topology

Although they do not state it in this special form, let us state the above mentioned special case of Pan and Zhu's result formally.

Theorem 2 (Pan and Zhu [7]) *For every even value of $k > 0$ and $r \geq 1$ we have*

$$\chi(M_r(\overline{C_{2k+1}})) = k + 2 = \chi(\overline{C_{2k+1}}) + 1,$$

while for every odd $k > 1$ and large enough r we have

$$\chi(M_r(\overline{C_{2k+1}})) = k + 1 = \chi(\overline{C_{2k+1}}).$$

The proof of Stiebitz's result in [1] gives more than just the chromatic number of the iterated generalized Mycielskian of odd cycles. It is based on the topological method to bound the chromatic number from below introduced by Lovász in his seminal paper [4]. The proof in [1] is based on a lemma (Lemma 3.1 in [1], see also as Theorem 5.9.6 in [5]) which implies that if the topological lower bound on the chromatic number provided by Lovász's method is t for a graph G then it is $t + 1$ for $M_r(G)$ for every r. This means that whenever this lower bound is tight for G, then we must have $\chi(M_r(G)) \geq \chi(G) + 1$ that can hold only with equality (implying that $t + 1$ will also be a tight lower bound for $\chi(M_r(G))$).

Graphs for which (a certain version of) the topological lower bound on their chromatic number is t are called *topologically t-chromatic* in [8]. The observation mentioned in the Introduction and the facts mentioned in the previous paragraph suggested that we should be able to prove that the graph $\overline{C_{2k+1}}$ is topologically $(k+1)$-chromatic if and only if k is even. Towards proving this the main observation was a trivial one: $\overline{C_{2k+1}}$ (for $k \geq 2$) is isomorphic to the circular (also called rational, cf. [2]) complete graph $K_{(2k+1)/2}$ that we define next.

Definition 3 *The circular complete graph $K_{p/q}$ is defined for positive integers $p \geq 2q$ as follows.*

$$V(K_{p/q}) = \{0, 1, \ldots, p-1\};$$

$$E(K_{p/q}) = \{\{i, j\} \colon q \leq |i-j| \leq p-q\}.$$

The name circular complete graph refers to the popular coloring parameter called circular chromatic number that can be defined as

$$\chi_c(G) := \inf\left\{\frac{p}{q} \colon G \to K_{p/q}\right\},$$

where $F \to H$ denotes the existence of a graph homomorphism from F to H (that is an edge-preserving map from $V(F)$ to $V(H)$). It is well-known that

$$\chi(G) - 1 < \chi_c(G) \leq \chi(G)$$

holds for any graph G, in particular, $\chi(K_{p/q}) = \left\lceil \frac{p}{q} \right\rceil$. For more about graph homomorphisms and the circular chromatic number we refer to [2, 15, 16].

In Subsection 3.3.4 of [9] the last two authors already listed the odd-chromatic circular complete graphs $K_{p/q}$ among those graphs G that are topologically $\chi(G)$-chromatic. As also explained there, this follows from the monotonicity of the topological lower bound of the chromatic number for graph homomorphism and the fact that the circular chromatic number of certain odd-chromatic topologically $\chi(G)$-chromatic graphs can be arbitrarily close to the lower bound $\chi(G)-1$. The first example found for such a family was that of generalized Mycielskians of complete graphs by Lam, Lin, Gu and Song [3]. On the other hand, if $\chi(G)$ is even, then topologically $\chi(G)$-chromatic graphs G always have $\chi_c(G) = \chi(G)$ as shown in [8] (and independently for the important special case of Schrijver graphs also by Meunier in [6]). This latter fact implies that even-chromatic circular complete graphs $K_{p/q}$ with non-integral p/q will not have equality between their chromatic number and its topological lower bound. Indeed, if there was equality for such p/q then $\chi_c(K_{p/q}) = \chi(K_{p/q}) = \lceil p/q \rceil > p/q$ would follow, an obvious contradiction.

The proof of the first statement in Theorem 2 already follows from the above: If $k > 0$ is even then $\left\lceil \frac{2k+1}{2} \right\rceil = k+1$ is odd implying that $\overline{C_{2k+1}} \cong K_{(2k+1)/2}$ is odd-chromatic therefore topologically t-chromatic with $t = k+1 = \chi(\overline{C_{2k+1}})$. This implies (by Stiebitz's result) that $\chi(M_r(\overline{C_{2k+1}})) = \chi(\overline{C_{2k+1}}) + 1 = k+2$.

The proof of the second statement can be checked by finding the corresponding coloring that is not too difficult.

Once one realizes the above relations it is quite natural to ask, whether $\chi(M_r(K_{p/q})) = \chi(K_{p/q})$ always holds when r is sufficiently large and $\chi(K_{p/q}) = \left\lceil \frac{p}{q} \right\rceil$ is even (and p/q non-integral). It is not hard to check that the answer is yes. The result of Pan and Zhu [7] also covers this, but it is even more general (since they also consider multicolorings) but we do not state it in its full generality. What we really wanted to emphasize and make better known is the special case we stated as Theorem 2.

3 Acknowledgement

We thank a referee for spotting a small but disturbing mistake in the original version of this extended abstract.

References

[1] András Gyárfás, Tommy Jensen, Michael Stiebitz, On graphs with strongly independent color classes, *J. Graph Theory* **46** (2004), 1–14.

[2] Pavol Hell, Jaroslav Nešetřil, *Graphs and Homomorphisms*, Oxford University Press, New York, 2004.

[3] Peter Che Bor Lam, Wensong Lin, Guohua Gu, Zengmin Song , Circular chromatic number and a generalization of the construction of Mycielski, *J. Combin. Theory Ser. B,* **89** (2003), 195–205.

[4] László Lovász, Chromatic number, Kneser's conjecture and homotopy, *J. Combin. Theory Ser. A* **25** (1978), 319–324.

[5] Jiří Matoušek, *Using the Borsuk-Ulam Theorem, Lectures on Topological Methods in Combinatorics and Geometry*, 2nd corrected prining, Springer-Verlag, Berlin, Heidelberg, 2008.

[6] Frédéric Meunier, A topological lower bound for the circular chromatic number of Schrijver graphs, *J. Graph Theory* **49** (2005), 257–261.

[7] Zishi Pan and Xuding Zhu, Multiple coloring of cone graphs, *SIAM J. Discrete Math.* **24** (2010), 1515–1526.

[8] Gábor Simonyi and Gábor Tardos, Local chromatic number, Ky Fan's theorem, and circular colorings, *Combinatorica* **26** (2006), 587–626.

[9] Gábor Simonyi and Gábor Tardos, Colorful subgraphs in Kneser-like graphs, *European J. Combin.* **28** (2007), 2188–2200.

[10] Michael Stiebitz, Beiträge zur Theorie der färbungskritischen Graphen, *Habilitation, TH Ilmenau* (1985)

[11] Claude Tardif, Fractional chromatic numbers of cones over graphs, *J. Graph Theory* **38** (2001), 87–94.

[12] Claude Tardif, The chromatic number of the product of 14-chromatic graphs can be 13, *Combinatorica* **42** (2022), 301–308.

[13] Claude Tardif, The chromatic number of the product of 5-chromatic graphs can be 4, *manuscript*, available at `https://www.researchgate.net/publication/365650263_THE_CHROMATIC_NUMBER_OF_THE_PRODUCT_OF_5-CHROMATIC_GRAPHS_CAN_BE_4`

[14] Marcin Wrochna, Smaller counterexamples to Hedetniemi's conjecture, arXiv:2012.13558 [math.CO].

[15] Xuding Zhu, Circular chromatic number: a survey, *Discrete Math.* **229** (2001), 371–410.

[16] Xuding Zhu, Recent Developments in Circular Colouring of Graphs, in: *Topics in Discrete Mathematics*, (Martin Klazar, Jan Kratochvíl, Martin Loebl, Jiří Matoušek, Pavel Valtr, Robin Thomas eds.), Algorithms and Combinatorics, vol. 26 (2006), 497–550.

Connecting Multicut and Multiway Cut using the Complement of the Demand Graph

Tamás Király[1]

Department of Operations Research
ELKH-ELTE Egerváry Research Group
Eötvös Loránd University
Budapest, Hungary
tamas.kiraly@ttk.elte.hu

Daniel P. Szabo

Department of Operations Research
Eötvös Loránd University
Budapest, Hungary
dszabo2@wisc.edu

Abstract: The Multiway Cut (MWC) problem asks for a minimum cut separating any two terminals from a given terminal set, while the Multicut (MC) problem requires terminal pairs specified by a demand graph H to be separated. Accurate approximation algorithms exist for MWC, while MC cannot be approximated to within a constant factor assuming the Unique Games Conjecture (UGC). We exhibit some characteristics of $\bar{H}$, the complement of the demand graph, where Multicut is equivalent to some Multiway Cut problem. One application we present concerns the solvability of MC on graphs of bounded treewidth $tw(\cdot)$. In particular, MC is APX-hard even when $tw(G \setminus V(H))$ and $tw(\bar{H})$ are bounded, yet fixed parameter tractable (FPT) in $tw(G)$ when the size of the maximum complete graph in $\bar{H}$ is bounded.

Keywords: graph connectivity; multicut; FPT; treewidth

1 Introduction

Graph cut problems have a long history in combinatorial optimization, with applications in network reliability, chip design, and image processing. The MULTIWAY CUT problem asks, given an input graph $G = (V, E)$ and a terminal set $S = \{s_1, \dots s_k\} \subseteq V$, for the minimum cut $C \subseteq E$ that separates each $s_i \in S$ into different components. As cuts can be viewed as partitions, this is equivalent to a node coloring of G with k colors such that each terminal s_i is colored with color i. We then seek to minimize the total weight of bichromatic edges.

The MULTICUT problem involves, in addition to a terminal set, a demand graph H on the set of terminal nodes, and asks for a minimum weight cut that disconnects each pair of nodes in $E(H)$. We assume that H contains no isolated nodes. A similar problem is MULTI-MULTIWAY CUT, where we are given a set of (not necessarily disjoint) terminal sets $S_1, S_2, \dots, S_c$ and seek to separate each pair $s, s' \in S_i$ with $s \neq s'$ for $i = 1 \dots c$. This generalizes MULTIWAY CUT when $c = 1$, and MULTICUT when each terminal set corresponds to an edge of H. The reverse can be said for multicuts, as MULTI-MULTIWAY CUT can be represented with a demand graph that is the union of c complete graphs. The key difference is in the parameters the problems are given.

The MULTIWAY CUT problem is NP-hard even for $k = 3$ [7]. Nonetheless, approximating MULTIWAY CUT has seen a lot of progress in the last decade, with improved analysis of the CKR relaxation [6] to achieve an approximation factor of 1.2965 [11], with the best known lower bound being slightly above 1.2 [3]. The MULTIWAY CUT problem can sometimes be solved exactly, depending on G. If G is a tree, a simple dynamic programming algorithm can compute the optimal multiway cut [9]. This extends to when G

[1]Research is supported by supported by the Lendület Programme of the Hungarian Academy of Sciences – grant number LP2021-1/2021, and by the National Research, Development and Innovation Fund of Hungary, financed under the ELTE TKP 2021-NKTA-62 funding scheme (project K10602/21).

has bounded treewidth $tw(G)$, for which [8] gives an FPT algorithm for MULTI-MULTIWAY CUT when c is constant.

The MULTICUT problem encodes MULTIWAY CUT when H is a complete graph, and thus is at least as difficult. It admits an $O(\log |E(H)|)$ approximation algorithm, and is super constant-hard under the Unique Games Conjecture (UGC) [4]. Nonetheless there are some conditions on G and H which make the problem tractable. As we saw, MULTICUT is even hard for general G when H is a triangle, which leaves little hope for conditions on H. One direction is when G is planar. Although it is still NP-hard, there is a polynomial time approximation scheme (PTAS) [1] for MULTIWAY CUT, and constant factor approximation algorithms for MULTICUT [12]. Another direction is to focus on when G resembles a tree. We already mentioned MULTIWAY CUT can be computed exactly when $tw(G \setminus S)$ is bounded. *Directed* MULTICUT can also be computed exactly when G is a tree, but it is hard for undirected trees. The next step is to look at fixed parameter tractability. When $tw(G + H)$ is bounded, where $G + H$ is just the union of the edges of G and H, the problem can be solved in polynomial time [10]. It is shown in [2] that if only $tw(G)$ and $tw(H)$ are separately bounded, then MULTICUT remains NP-hard. This is because MULTICUT in stars is equivalent to MULTICUT in trees of height 2 with H composed only having vertex disjoint edges. They also show it is FPT in $tw(G + \bar{H})$ using a dynamic programming algorithm, where $\bar{H}$ is the complement graph of H. One of our motivating questions is what the complexity of MULTICUT is when $tw(G)$ and $tw(\bar{H})$ are separately bounded. We will show that the problem remains hard when only $tw(\bar{H})$ and $tw(G \setminus V(H))$ is bounded, as in the Erdős-Székely result for MULTIWAY CUT [9], but can be solved in polynomial time if $tw(G)$ itself is bounded as well.

A powerful result on approximability based on the properties of H is given by Chekuri and Madan [5]. They give a 2-approximation when H excludes an induced matching of size t for constant t. This is equivalent to all $\bar{H}$ that do not have an induced near complete $K_t \setminus tK_2$, which is true for any $\bar{H}$ with bounded degree. They go on to ask what role the demand graph plays in the approximability of MULTICUT, and when the approximation factors of MULTIWAY CUT can be used in this setting.

We give hardness results for the case where the treewidth of $\bar{H}$ is bounded and $G \setminus S$ is a forest. We also give algorithms FPT in $tw(G \setminus S)$ when $\bar{H}$ is the disjoint union of complete graphs, or $|E(\bar{H})|$ is bounded, or $\bar{H}$ is the disjoint union of complete graphs along with a constant number of additional edges. We also give an FPT algorithm in $tw(G)$ when the size of the maximum stable set in H is bounded. These results are summarized in Table 1.

Parameters	Constraints	Complexity
	$tw(G + \bar{H})$ bounded	P (Theorem 12 in [2])
$tw(G \setminus V(H))$, $tw(\bar{H})$		APX-hard (Theorem 7)
$tw(G \setminus V(H))$	$\bar{H} = \bigsqcup_i K_{n_i}$	FPT (Theorem 1)
$tw(G \setminus V(H))$, $\lvert E(H)\rvert$		FPT (Theorem 3)
$tw(G \setminus V(H))$, $\lvert E(H')\rvert$	$\bar{H} = \bigsqcup_i K_{n_i} + H'$	FPT (Theorem 4)
$tw(G)$	$\alpha(H)$ bounded	FPT(Theorem 6)

Table 1: A summary of the results on the complexity of MC using $\bar{H}$. Here $\sqcup$ is the disjoint set union, and $\alpha(H)$ is the size of the maximum stable set in H, and H' is an arbitrary graph. All of the parameters include the graph G and the demand graph H.

Our main tool is the simple fact that any multicut solution creates some partition of $V(G)$, and in any component the terminals that appear must induce a complete graph in $\bar{H}$. This works in both directions, as any covering of the vertices of $\bar{H}$ by complete graphs corresponds to a multiway cut instance. Thus, if we can grasp the possible partitions into complete graphs of $\bar{H}$, we can reduce the complexity to the multiway cut problem.

2 Tractable Cases

2.1 Treewidth

We first define a tree decomposition of a graph G. A *tree decomposition* of $G = (V, E)$ is a pair $(\{X_i\}_{i \in I}, T)$ of bags X_i indexed by I and a tree T with vertex set I that satisfies the following properties:

- $\bigcup_{i \in I} X_i = V$.
- Each edge u, v in E is contained in some X_i.
- For each triple $i, j, \ell \in I$, if j lies on the path between i and ℓ, then $X_i \cap X_\ell \subseteq X_j$.

The width of a given tree decomposition is $\max_{i \in I} |X_i| - 1$. The treewidth $tw(G)$ is the minimum width of any tree decomposition, and can be thought of as a measure of how close G is to a tree.

2.2 Basic Applications

We begin by going through some of the connections between MULTICUT and MULTIWAY CUT when we can enumerate all possible vertex clique covers of $\bar{H}$. A vertex clique cover is a partition of the vertices of $\bar{H}$ such that each partition induces a complete subgraph.

Theorem 1 *The* MULTICUT *problem is FPT in $tw(G \setminus V(H))$ when $\bar{H}$ is the disjoint union of complete graphs.*

PROOF: Let (G, H) be a MULTICUT instance with $\bar{H} = \bigsqcup_{i=1}^{k} K_{n_i}$. Create an instance (G', S) for MULTIWAY CUT where G' is formed by merging K_{n_i} into one vertex s_i for each i, and $S = \{s_1, \ldots, s_k\}$. We can solve MULTIWAY CUT in polynomial time on bounded $tw(G' \setminus S)$ [8]. When $tw(G \setminus V(H))$ is bounded, so is $tw(G' \setminus S)$, and the solution to the MULTIWAY CUT instance is the same as that of the MULTICUT instance. □

Theorem 2 *The* MULTICUT *problem is FPT in $tw(G \setminus V(H))$ and $|E(\bar{H})|$.*

PROOF: Consider any clique cover of $\bar{H}$ by disjoint complete subgraphs. By Theorem 1, we can solve MULTICUT in polynomial time on the instance where the complement of the demand graph is this clique covering. The number of possible such decompositions is no greater than the number of ways to partition the edge set, which is a function of $|E(\bar{H})|$. Thus we can look at each partition, and take the minimum MULTIWAY CUT as our solution. □

A useful corollary is that MULTICUT can be solved using MULTIWAY CUT even when $|E(H)|$ is bounded. The reason is simply that $|E(\bar{H})|$ is bounded by a function of $|E(H)|$ since we assumed that no terminal node is isolated in H. Specifically $|E(\bar{H})| \leq |V(H)|^2 \leq (2|E(H)|)^2$, since we assumed that no terminal node is isolated in H.

Corollary 3 *The* MULTICUT *problem is FPT in $tw(G \setminus V(H))$ and $|E(H)|$.*

Finally we combine Theorems 1 and 2.

Theorem 4 *The* MULTICUT *problem is FPT in $tw(G \setminus V(H))$ and $|E(H')|$ when $\bar{H} = \bigsqcup_i K_{n_i} + H'$.*

PROOF: We can use the proof method of Theorem 2 to bound the number of vertex clique covers. The decomposition from Theorem 1 is one such cover, and any other decomposition would include a partition of some nonempty subset of H'. Furthermore, this partition and subset determines the cliques in the

cover. Thus, the number of clique covers is no greater than the total number of possible partitions of subsets of $E(H')$, which is still bounded by a function of $|E(H')|$. □

Finally we mention that all of the above results, as well as Theorem 6, reduce the MULTICUT instance to a MULTIWAY CUT instance, and therefore can be approximated well [11]. We summarize this in the following corollary:

Corollary 5 *There is a 1.2965-approximation algorithm for* MULTICUT *when H satisfies any of the following:*

- $\bar{H} = \bigsqcup_i K_{n_i} + H'$ *for some H' with a constant number of edges.*
- *The size of the maximum stable set $\alpha(H)$ is bounded.*

2.3 Dynamic Programming Algorithm when G is a Tree

In this section we give an FPT algorithm in $tw(G)$ when $\alpha(H)$ is bounded, where $\alpha(\cdot)$ denotes the size of the maximum independent set. The maximum independent set in H corresponds to the largest complete graph in $\bar{H}$. If the size of the largest complete graph in $\bar{H}$ is bounded, the number of ways to put terminals in a single component is polynomial in n, and we can build these components up as we traverse the tree bottom-up.

In this article, we describe the algorithm only for the case when G is a tree, but it can be generalized to bounded treewidth via the usual methods.

Define $n = |V(G)|$, let $S = V(H)$ be the terminal set, and let $\alpha(H) = k$. Root G at an arbitrary vertex r. We can assume each terminal s is a leaf in G, and no other nodes are leaves. Indeed, any terminal s can be made a degree 1 node in G by adding a dummy vertex v to replace s, and an edge (v, s) of arbitrary large weight, and any nonterminal leaf can be removed without changing the problem. Define the set of all possible subsets of the terminals that can be in a single component (all complete graphs in $\bar{H}$) as K. Note that $|K| \leq \binom{n}{k}$. For a nonleaf node u in G, the subproblem $\textsc{TreeCut}(u, T)$ is parameterized by some terminals $T \in K$ and stores the optimal cut for the subtree under u given that the terminals in T are in the component of u. If u has children $v_1, \ldots, v_l$ this can be calculated as

$$\textsc{TreeCut}(u, T) = \sum_{j=1}^{l} \min \left\{ \min_{T' \in K : T' \cap T = \emptyset} \textsc{TreeCut}(v_j, T') + w(u, v_j), \textsc{TreeCut}(v_j, T) \right\}.$$

The output is then the minimum value at the root, namely $\min_{T \in K} \textsc{TreeCut}(r, T)$. This is written explicitly in Algorithm 1. Although the algorithm as written here only finds the value of the minimum multicut, backtracing can give the edges as well.

Theorem 6 *Algorithm 1 outputs the cost minimum multicut in polynomial time.*

PROOF: Correctness follows from induction on the depth of u. Each leaf node is a terminal vertex that only has a value less than ∞ when it is in it's own component. For a given nonleaf node u with children $v_1, \ldots, v_l$ and some $T \in K$ that can be in a single component of a valid cut, if u is in the same component as the terminals in T, each child v_j is either in this component as well or in some other one. If v_j is in another component T', T' cannot have any of the same terminals as T and the edge (u, v_j) must be cut. The optimal cost is then just the sum of the minimum of these *independent* choices for each v_j. Note that these choices are only independent because G itself is a tree, rather than just $G \setminus S$.

The running time of a naïve implementation of the algorithm is $O(n|K|^2)$, which is $O(n^{2k+1})$ and polynomial time when k is bounded. □

When G only has bounded treewidth, we can generalize this algorithm by adding another dimension for how a bag is partitioned, and what terminals are in the same component as each part of the partition. This would add, for some exponential unary function f and polynomial (for fixed $\alpha(H)$) binary function g, an additional factor of $f(tw(G)) \cdot g(n, tw(G))$ to the running time. We omit the details for brevity.

Algorithm 1 A dynamic programming algorithm for MULTICUT when G is a tree and H has a maximal independent set of bounded size

Require: G a tree rooted at r, $\alpha(H)$ bounded.
  $n \leftarrow |V|$, $k \leftarrow \alpha(H)$, $K \leftarrow \{T \subseteq S : |T| \leq k, T \text{ induces a complete graph in } \bar{H}\}$.
  for all $s \in S, T \in K$ **do**

$$\textsc{TreeCut}(s,T) \leftarrow \begin{cases} 0 \text{ if } s \in T \\ \infty \text{ otherwise} \end{cases}$$

  end for
  Let $u_1, u_2, \ldots, u_m$ be a bottom up traversal of the interior nodes of T.
  for $i = 1$ to m **do**
    Let $\{v_1, \ldots, v_l\}$ be the children of u_i.
    for all $T \in K$ **do**

$$\textsc{TreeCut}(u,T) \leftarrow \sum_{j=1}^{l} \min \begin{cases} \min\limits_{T' \in K : T' \cap T = \emptyset} \textsc{TreeCut}(v_j, T') + w(u, v_j) \\ \textsc{TreeCut}(v_j, T) \end{cases}$$

    end for
  end for
  return $\min_{T \in K} \textsc{TreeCut}(r,T)$.

3 Hardness of Bounded $tw(\bar{H}), tw(G \setminus V(H))$

We reduce the vertex cover problem to our problem. We will use a demand graph where $\bar{H}$ is a set of disjoint 2-paths. In this case, cuts in the constructed graph G' correspond to matchings in $\bar{H}$ (the pairs of non-separated terminals), of which there are exponentially many. These will correspond to vertex covers of G, with weights chosen such that the minimum weight multicut is the same as the minimum weight vertex cover.

Theorem 7 *The* MULTICUT *problem is APX-hard with bounded* $tw(\bar{H})$ *and* $G \setminus V(H)$ *a forest.*

PROOF: Let $G = (V, E)$ be an input to VERTEX-COVER. Let H, the demand graph of the MULTICUT instance, have vertices corresponding to 3 copies of V: V_1, V_2, and V_3. Let $\bar{H}$ have edge set $\{(v_1, v_2), (v_1, v_3) : v \in V\}$. Construct the MULTICUT instance (G', H) by replacing each edge in G with the gadget shown in Figure 1, with weights as shown for some $\lambda > 1$ arbitrarily large. For each vertex $v \in V$ there is only one weight 1 edge going to v_3, but there may be multiple instances of v_2, one for each edge. This is achieved by adding an extra vertex for each repeated instance of v_2 with an edge of infinite weight connected to the true terminal node v_2. An example for the triangle graph is given in Figure 2.

Thus $G \setminus V(H)$ consists only of isolated vertices, and has treewidth zero. Any multicut of G' must cut at least two of the edges adjacent to any nonterminal node. Any minimum multicut of G' cuts at most three of these edges.

The mapping between multicuts of G' and vertex covers of G covers $v \in V$ whenever v_1 and v_2 are in the same component. In this case, each nonterminal edge neighboring v in G' only needs 2 edges cut. For λ large, any minimum multicut will cut 2 edges in G' for each edge in G, plus a cover C of E, incurring a cost of $2\lambda|E| + |C|$. This is minimal if and only if C is a minimum vertex cover C^*, so the minimal multicut of G' corresponds the minimal vertex cover of G. □

References

[1] MH. BATENI, MT. HAJIAGHAYI, P. KLEIN, M. CLAIRE, A Polynomial-time Approximation Scheme for Planar Multiway Cut, *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms* (2012) 639–655

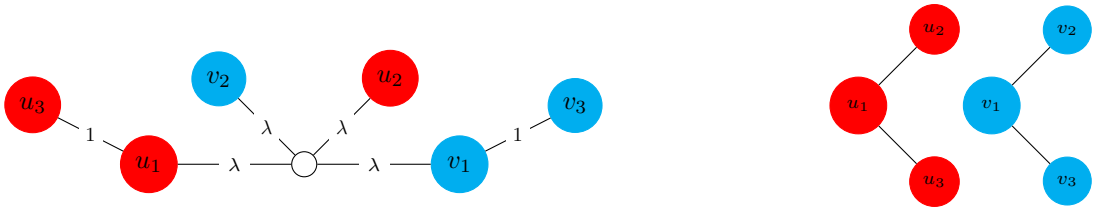


Figure 1: The gadget used to connect multicuts to vertex covers. On the left is the graph G, and on the right the complement of the demand graph $\bar{H}$.

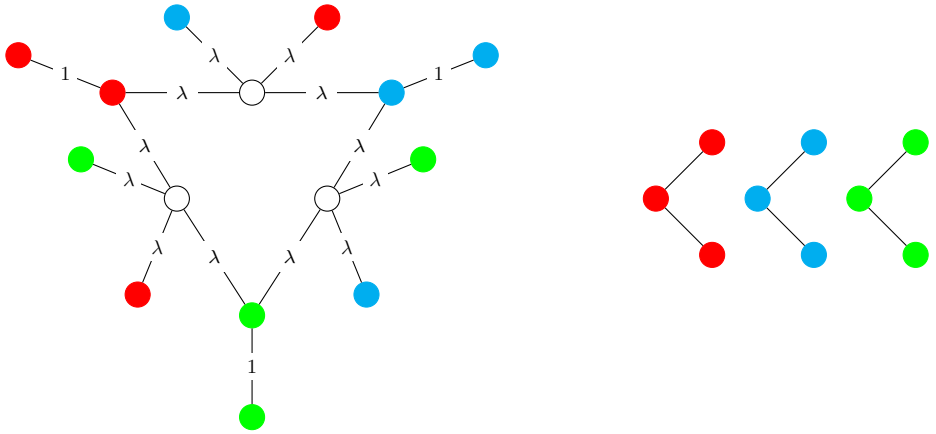


Figure 2: An example of the reduction on a triangle graph

[2] C. Bentz, P. Le Bodic, Complexity of the multicut problem, in its vanilla, partial and generalized versions, in graphs of bounded treewidth, *Theoretical Computer Science* **809** (2020) 239–249

[3] K. Bérczi, K. Chandrasekaran, T. Király, V. Madan, Improving the integrality gap for multiway cut, *Mathematical Programming* **183** (2020) 171–193

[4] S. Chawla, R. Krauthgamer, R. Kumar, Y. Rabani, D. Sivakumar, On the hardness of approximating multicut and sparsest-cut, *Comput. Complex.* **15** (2006) 94–114

[5] C. Chekuri, V. Madan, Approximating multicut and the demand graph, *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms* (2017) 855–874

[6] G. Călinescu, H. Karloff, Y. Rabani, An Improved Approximation Algorithm for MULTIWAY CUT, *Journal of Computer and System Sciences* **60** (2000) 564–574

[7] E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, M. Yannakakis, The Complexity of Multiterminal Cuts, *SIAM Journal on Computing* **23** (1994) 864–894

[8] X. Deng, B. Lin, C. Zhang, Multi-Multiway Cut Problem on Graphs of Bounded Branch Width, *Frontiers in Algorithmics and Algorithmic Aspects in Information and Management* (2013) 315–324

[9] P.L. Erdős, A. Frank, L. Székely, Minimum multiway cuts in trees, *Discrete Applied Mathematics* **87** (1998) 67–75

[10] G. Gottlob, S.T. Lee, A logical approach to multicut problems, *Information Processing Letters* **103** (2007) 136–141

[11] A. Sharma, J. Vondrák, Multiway cut, pairwise realizable distributions, and descending thresholds, *Proceedings of the forty-sixth annual ACM symposium on Theory of computing* (2014) 724–733

[12] E. Tardos, V.V. Vazirani, Improved bounds for the max-flow min-multicut ratio for planar and Kr,r-free graphs, *Information Processing Letters* **47** (1993) 77–80

Submodular flows with minimal spread[1]

Alpár Jüttner

Department of Operations Research
Eötvös Loránd University
Pázmány Péter stny. 1/C, Budapest, Hungary
alpar.juttner@gmail.com

Eszter Szabó

Department of Operations Research
Eötvös Loránd University
Pázmány Péter stny. 1/C, Budapest, Hungary
szeti97@gmail.com

Abstract: Balanced optimization problems aim to find the most equitable distribution of resources. This paper examines the Balanced Submodular Flow Problem, that is the problem of finding a feasible submodular flow minimizing the difference between flow values along the arcs. An algorithm is presented to solve it with $\mathcal{O}(m^4)$ submodular function minimizations. We also show a strongly polynomial algorithm for the weighted case.

Keywords: submodular flow, balanced optimization, Handler-Zang Method

1 Introduction

In balanced optimization problems, the aim is to find the most equitable distribution of resources. Several problems have been analysed in the literature such as the balanced spanning tree problem which has been studied by Longshu Wu [9]. Another example is the balanced assignment problem by Martello[7]. Ahuja proposed a parametric simplex method for the general balanced linear programming problem [8]. Punnen et al. introduced a strongly NP-hard problem, which is called the quadratic balanced optimization problem, and showed some algorithms in a special case.

Scutella investigated Balanced Network Flow Problem [1],[10]. The problem is to find a feasible flow minimizing the difference between the maximum and minimum flow values along the arcs, i.e. $\max_{e\in A} f(e) - \min_{e\in A} f(e)$. Scutella presented an algorithm using Newton's approach by performing $\mathcal{O}(n^2 \log^3(n))$ max-flow computations, where n is the number of the nodes.

This paper examines a similar problem with submodular flows. The problem aims to find a feasible submodular flow minimizing the difference between flow values along the arcs. A strongly polynomial algorithm is presented to solve Balanced Submodular Flow Problem with $\mathcal{O}(m^4)$ submodular function minimizations, where m is the number of edges in the input graph. In section 4, the Balanced Integral Submodular Flow Problem is introduced and the optimal integral solution is given using the fractional optimum. Finally, section 5 describes the Weighted Balanced Submodular Flow problem, that is the problem of finding a feasible submodular flow minimizing the difference between the maximum and minimum weighted flow values along the arcs. For a given graph $G = (V, A)$ and any $c : A \longrightarrow \mathbb{R}^+$, the problem aim to find a feasible submodular flow f minimizing $\max_{e\in A} c(e)f(e) - \min_{e\in A} c(e)f(e)$. We show that the Handler-Zang method solves this problem in $\mathcal{O}(n^4m^6 \log^6(m))$ number of iterations.

2 Preliminaries

Definition 1 *For an underlying set V, the set functions $b, p : \mathcal{P}(V) \longrightarrow \mathbb{R}$ are called* submodular *and* supermodular *if*

$$b(X) + b(Y) \geq b(X \cup Y) + b(X \cap Y) \tag{1}$$

[1]Supported by ELKH-ELTE Egerváry Research Group and MTA-ELTE Momentum Matroid Optimization Research Group

and

$$p(X) + p(Y) \leq p(X \cup Y) + p(X \cap Y) \tag{2}$$

holds for each subsets $X, Y \subseteq V$, respectively. A functions is called modular *if it is both sub- and supermodular.*

Theorem 2 (Orlin [5]) *Assuming that the value of a submodular function b can be computed for any $X \subseteq V$ in time T, then $\min\{b(X) : X \subseteq V\}$ can be computed in time $\mathcal{O}(n^5T + n^6)$*

Definition 3 *For a directed graph $G = (V, A)$ and a subset of vertices $X \subseteq V$, let $\tilde{\varrho}(X)$ and $\tilde{\delta}(X)$ denote the set of arcs entering X and leaving X, respectively. For a vector $x \in \mathbb{R}^A$, let*

$$\varrho_x(X) := \sum_{e \in \tilde{\varrho}(X)} x(e), \quad \delta_x(X) := \sum_{e \in \tilde{\delta}(X)} x(e) \quad \text{and} \quad \partial_x(X) := \varrho_x(X) - \delta_x(X). \tag{3}$$

Furthermore, let $\varrho(X)$, $\delta(X)$ and $\partial(X)$ denote the number of edges entering X, leaving X, and their difference, respectively.

A G is called Eulerian *if $\partial(v) = 0$ for all $v \in V$. Note that G is not required to be connected.*

It is straightforward to check that $\varrho_x(X)$ and $\delta_x(X)$ are submodular functions for any nonnegative vector x. If $l, u \in \mathbb{R}^A$ and $l \leq u$, then $\varrho_u(X) - \delta_l(X)$ is submodular and $\varrho_l(X) - \delta_u(X)$ is supermodular.

Definition 4 *Let us given a directed graph $G = (V, A)$ and a submodular function $b : \mathcal{P}(V) \longrightarrow \mathbb{R}$. A vector $x \in \mathbb{R}^A$ is called a* submodular flow *if*

$$\varrho_x(X) - \delta_x(X) \leq b(X) \tag{4}$$

holds for each $X \subseteq V$.

For vectors $l, u \in \mathbb{R}^A$, a submodular flow x is called (l, u)-bounded *if $l \leq x \leq u$.*

Theorem 5 *For lower and upper bounds $l, u \in \mathbb{R}^A$, there exists an (l, u)-bounded submodular flow if and only if $l \leq u$ and*

$$\varrho_l(X) - \delta_u(X) \leq b(X) \tag{5}$$

hold for each $X \subseteq V$.

Theorem 6 (Frank, [3]) *Assuming that the value of a submodular function b can be computed for any $X \subseteq V$ in time T, then a feasible submodular flow can be found in $\mathcal{O}(n^5T)$ time.*

3 Balanced Submodular Flows

Definition 7 *The* spread $\sigma(x)$ *of a vector $x \in \mathbb{R}^A$ is the value*

$$\max_{a \in A} x(a) - \min_{a \in A} x(a)$$

The *Balanced Submodular Flow Problem* is to find a submodular flow of minimum spread.

Problem 8 *For a given directed graph $G = (V, A)$ and a submodular function $b : \mathcal{P}(V) \longrightarrow \mathbb{Z}$, the aim is to find a submodular flow of minimum spread, i.e find*

$$\sigma^* := \min\{\sigma(x) : \varrho_x(X) - \delta_x(X) \leq b(X) \quad \forall X \subseteq V\} \tag{6}$$

along with a minimizing vector x^*.

Definition 9 *For an arbitrary real value $\kappa \in \mathbb{R}$, let $s(\kappa)$ denote the minimal value σ for which there exists a $(\kappa\mathbb{1}, (\kappa + \sigma)\mathbb{1})$-bounded submodular flow x.*

With this notation

$$\sigma^* := \min_{\kappa \in \mathbb{R}} s(\kappa) \tag{7}$$

From Theorem 5 it follows that

$$s(\kappa) = \min \{\sigma \geq 0 : 0 \leq b(X) + \sigma\delta(X) - \kappa\partial(X) \quad \forall X \subseteq V\} \tag{8}$$

Claim 10 *$s(\kappa)$ is a convex function.*

PROOF: For any $\kappa_1, \kappa_2 \in \mathbb{R}$, let x_1 and x_2 be submodular flows such that $\kappa_i \mathbb{1} \leq x_i \leq (\kappa_i + s(\kappa_i))\mathbb{1}$, and let $0 \leq \lambda \leq 1$. Then $x' := \lambda x_1 + (1-\lambda)x_2$ is also a submodular flow and $[\lambda\kappa_1 + (1-\lambda)\kappa_2]\mathbb{1} \leq x' \leq [\lambda(\kappa_1 + s(\kappa_1)) + (1-\lambda)(\kappa_2 + s(\kappa_2))]\mathbb{1}$, therefore

$$s(\lambda\kappa_1 + (1-\lambda)\kappa_2) \leq \lambda s(\kappa_1) + (1-\lambda)s(\kappa_2) \tag{9}$$

□

In the following, a dual characterization of the value of the minimum spread and algorithms for the Balanced Submodular Flow Problem will be given. For some technical reasons, the case of Eulerian and non-Eulerian graphs are treated separately.

3.1 Eulerian graphs

Clearly, if G is Eulerain and $x \in \mathbb{R}^A$ is a submodular flow, then $x + c\mathbb{1}$ is also a submodular flow for any $c \in \mathbb{R}$. Therefore the Balanced Submodular Flow problem reduces to the problem of finding the minimum value σ^* for which there exists a submodular flow $0 \leq x^* \leq \sigma^*\mathbb{1}$. Applying Theorem 5, σ^* is the smallest value for which $b(X) + \sigma^*\delta(X) \geq 0$ holds for all $X \subseteq V$. In other words, we are looking for the root of the function

$$f(\sigma) := \min\{b(X) + \sigma\delta(X) : X \subseteq V\} \tag{10}$$

This immediately gives the following dual characterization of the minimum spread submodular flows.

Theorem 11 *Assume that G is Eulerian. Then*

$$\sigma^* = \max\left\{\frac{-b(X)}{\delta(X)} : X \subseteq V, \delta(X) > 0\right\} \tag{11}$$

Therefore the problem reduces to a fractional optimization problem, the optimum of which can be calculated using the standard Newton-Dinkelbach procedure[12], which is outlined in Algorithm 1. It is straightforward to see that $\delta(X_i)$ is strictly decreasing in every iteration and a standard argument shows that the final set X_i indeed maximizes the value $\frac{-b(X)}{\delta(X)}$, thus

Theorem 12 *Algorithm 1 finds the value σ^* of the minimum spread and the corresponding dual X after at most m iterations.*

3.2 Non-Eulerian Graphs

Theorem 13 *Assume that G is not Eulerian. Then*

$$\sigma^* = \max\left\{\frac{b(X)\partial(Y) - b(Y)\partial(X)}{\delta(Y)\partial(X) - \delta(X)\partial(Y)} : X, Y \subseteq V, \partial(X) \geq 0, \partial(Y) < 0\right\}, \tag{12}$$

and σ^* along with the maximizing sets X and Y can be calculated in $\mathcal{O}(m^4T)$ time, where T denotes the time complexity of a submodular function minimization.

Algorithm 1 Minimum spread calculation in Eulerian graphs

1: Let $\sigma_1 := 0$
2: $i := 1$
3: **loop**
4: Let $X_i := \arg\min\{b(X) + \sigma_i\delta(X) : X \subseteq V\}$
5: **if** $b(X_i) + \sigma_i\delta(X_i) \geq 0$ **then**
6: RETURN σ_i, X_i
7: **else if** $\delta(X_i) = 0$ **then**
8: RETURN "INFEASIBE"
9: **else**
10: $\sigma_{i+1} := \frac{-b(X_i)}{\delta(X_i)}$
11: **end if**
12: $i \longleftarrow i + 1$
13: **end loop**

In order to prove the theorem above, we first show that the expression above constitutes a lower bound of σ^* for any pairs of sets X and Y, then give an algorithm for finding X^* and Y^* for which the equality holds.

Lemma 14 *Let $X, Y \subseteq V$ such that $\partial(X) \geq 0$ and $\partial(Y) < 0$ then*

$$\sigma^* \geq \frac{b(X)\partial(Y) - b(Y)\partial(X)}{\delta(Y)\partial(X) - \delta(X)\partial(Y)} \tag{13}$$

Proof: By definition of x^*, there exists a real value κ such that $\kappa\mathbb{1} \leq x^* \leq (\kappa + \sigma^*)\mathbb{1}$. Using Theorem 5 with $l := \kappa\mathbb{1}$ and $u := (\kappa + \sigma^*)\mathbb{1}$ we get that

$$\kappa\partial(X) - \sigma^*\delta(X) \leq b(X) \tag{14}$$

and

$$\kappa\partial(Y) - \sigma^*\delta(Y) \leq b(Y). \tag{15}$$

From which we get that

$$b(X)\partial(Y) + \sigma^*\delta(X)\partial(Y) \leq \kappa\partial(X)\partial(Y) \leq b(Y)\partial(X) + \sigma^*\delta(Y)\partial(X) \tag{16}$$

therefore

$$\sigma^*\big(\delta(Y)\partial(X) - \delta(X)\partial(Y)\big) \geq b(X)\partial(Y) - b(Y)\partial(X) \tag{17}$$

□

In order to finish the proof of Theorem 13, we give an algorithm that actually finds the pairs of sets X and Y for which 13 holds with equality. The procedure is outlined in Algorithm 2.

Computing C in line 6 involves in the minimizations of the submodular function $b(X) + \sigma_i\delta(X) - \kappa_i\partial(X)$. If the algorithm exits at line 8, then $\kappa_i\partial(X) - \sigma_i\delta(X) \leq b(X)$ holds for all $X \subseteq V$, therefore — by Theorem 5 — there exists a submodular flow $\kappa_i\mathbb{1} \leq x \leq (\kappa_i + \sigma_i)\mathbb{1}$. On the other hand, Theorem 14 ensures that the spread of any submodular flow is at least σ_i. If the algorithm exits at line 10, then both $\varrho(C)$ and $\delta(C)$ is zero, but $b(C) < 0$, thus the submodular flow problem has no feasible solution at all.

Note, that $\partial(Y_i) < 0$ in every iteration, thus σ_i and κ_i are valid, i. e. their denominator can't be zero.

It is straightforward to see that the value σ_i strictly instreases at each iteration, therefor the algorithms terminates after a finite number of iterations.

Algorithm 2 Minimum spread calculation in non-Eulerian graphs

Choose $X_1, Y_1 \subseteq V$ such that $\partial(X_1) > 0$ and $\partial(Y_1) < 0$
$i := 1$
loop
  $\sigma_i := \dfrac{b(X_i)\partial(Y_i) - b(Y_i)\partial(X_i)}{\delta(Y_i)\partial(X_i) - \delta(X_i)\partial(Y_i)}$
  $\kappa_i := \dfrac{b(X_i)\delta(Y_i) - b(Y_i)\delta(X_i)}{\delta(Y_i)\partial(X_i) - \delta(X_i)\partial(Y_i)}$
  $C := \arg\min\{b(X) + \sigma_i\delta(X) - \kappa_i\partial(X) : X \subseteq V\}$
  if $b(C) + \sigma_i\delta(C) - \kappa_i\partial(C) \geq 0$ **then**
    RETURN $\kappa_i, \sigma_i, X_i, Y_i$
  else if $\partial(C) = 0$ and $\delta(C) = 0$ **then**
    RETURN "INFEASIBE"
  else if $\partial(C) \geq 0$ **then**
    $X_{i+1} := C$
    $Y_{i+1} := Y_i$
  else
    $X_{i+1} := X_i$
    $Y_{i+1} := C$
  end if
  $i \longleftarrow i + 1$
end loop

3.2.1 Running time of the algorithm

In the following, it will be shown that not only Algorithm 2 is finite, but in fact it runs in a strongly polynomial time.

Let us consider two sets $Z_1, Z_2 \subseteq V$ of the *same type* if $\delta(Z_1) = \delta(Z_2)$, $\rho(Z_1) = \rho(Z_2)$ and $b(Z_1) = b(Z_2)$.

Theorem 15 *The algorithm can find at most $m^2 + m$ sets of different type.*

Proof: First, observe that if $\delta(X_i) > 0$, Equation 6 can be rewritten as follows

$$\arg\min\{b(X) + \sigma_i\delta(X) - \kappa_i\partial(X)\} = \arg\max\left\{\delta(x)\Big(\frac{\kappa_i\partial(X) - b(X)}{\delta(X)} - \sigma_i\Big)\right\} \tag{18}$$

Let us consider the subsets $X_{i_1}, X_{i_2}, \ldots, X_{i_k}$ found by the algorithm, such that

$$\frac{\partial(X_{i_1})}{\delta(X_{i_1})} = \frac{\partial(X_{i_2})}{\delta(X_{i_2})} = \cdots = \frac{\partial(X_{i_k})}{\delta(X_{i_k})}$$

where $\delta(X_{i_j}) > 0$. Let us assume that $\frac{-b(X_{i_1})}{\delta(X_{i_1})} < \frac{-b(X_{i_2})}{\delta(X_{i_2})} < \cdots < \frac{-b(X_{i_k})}{\delta(X_{i_k})}$. Note that if X, X' are found by the algorithm and $\frac{\partial(X)}{\delta(X)} = \frac{\partial(X')}{\delta(X')}$ and $\frac{-b(X)}{\delta(X)} = \frac{-b(X')}{\delta(X')}$, then $\delta(X) = \delta(X')$ must hold and they can be considered equivalent. For any κ, σ, we have that

$$\frac{\kappa\partial(X_{i_1}) - b(X_{i_1})}{\delta(X_{i_1})} - \sigma \leq \frac{\kappa\partial(X_{i_2}) - b(X_{i_2})}{\delta(X_{i_2})} - \sigma \leq \cdots \leq \frac{\kappa\partial(X_{i_k}) - b(X_{i_k})}{\delta(X_{i_k})} - \sigma$$

Because X_{i_1} maximizes the right hand side of Equation 18 for $\kappa_{i_1}, \sigma_{i_1}$, therefore $\delta(X_{i_1}) > \delta(X_{i_2})$. By the same token, we get that $\delta(X_{i_1}) > \delta(X_{i_2}) > \cdots > \delta(X_{i_k})$. Thus, $\partial(X)$ and $\partial(Y)$ must be different for any pair of subsets X, Y such that $\delta(X) = \delta(Y)$. So, the algorithm finds at most m subsets with a

particular value of $\delta(X)$. Since there are at most m different values of $\delta(X)$, the algorithm finds at most m^2 different sets, such that $\delta(X) > 0$.

Now, let $X_{j_1}, X_{j_2}, \ldots, X_{j_l}$ be the sets found by the algorithm such that $\delta(X_i) = 0$. We can assume that $\frac{b(X_{j_1})}{\rho(X_{j_1})} > \frac{b(X_{j_2})}{\rho(X_{j_2})} > \cdots > \frac{b(X_{j_l})}{\rho(X_{j_l})}$. Similarly to the above argument, weget that $\rho(X_{j_1}) < \rho(X_{j_2}) < \cdots < \rho(X_{j_l})$, which means that there are at most m different subsets with $\delta(X) = 0$. Therefore, the algorithm finds at most $m^2 + m$ different sets. □

In order to estimate the number of iterations of the algorithm, we give an upper bound on how many times a particular set C (or another one of the same type) can repeatedly appear in the sequences $X_1, X_2, \ldots$ and $Y_1, Y_2, \ldots$.

Lemma 16 *Let C be a set, that is found by the algorithm. Then C can be found at most $(\delta(C)+\rho(C))m$ times during iterations.*

Proof: Let $i_1 - 1, i_2 - 1, ..., i_k - 1$ be the iterations, where the algorithm found C.
Then $\sigma_{i_j} = \frac{\kappa_{i_j}\partial(C) - b(C)}{\delta(C)}$. By step 6, the following holds true:

$$b(C_{i_j}) + \sigma_{i_j}\delta(C_{i_j}) - \kappa_{i_j+1}\partial(C_{i_j+1}) \leq b(C_{i_{j+1}}) + \sigma_{i_j}\delta(C_{i_{j+1}}) - \kappa_{i_j}\partial(C_{i_{j+1}})$$
$$b(C_{i_{j+1}}) + \sigma_{i_{j+1}}\delta(C_{i_{j+1}}) - \kappa_{i_{j+1}}\partial(C_{i_{j+1}}) \leq b(C_{i_j}) + \sigma_{i_{j+1}+1}\delta(C_{i_j}) - \kappa_{i_{j+1}}\partial(C_{i_j})$$

This means that

$$(\sigma_{i_{j+1}} - \sigma_{i_j})\delta(C_{i_{j+1}}) + (\kappa_{i_j} - \kappa_{i_{j+1}})\partial(C_{i_{j+1}}) \leq (\sigma_{i_{j+1}} - \sigma_{i_j})\delta(C_{i_j}) + (\kappa_{i_j} - \kappa_{i_{j+1}})\partial(C_{i_j})$$
$$\delta(C_{i_{j+1}}) - \frac{\delta(C)}{\partial(C)}\partial(C_{i_{j+1}}) \leq \delta(C_{i_j}) - \frac{\delta(C)}{\partial(C)}\partial(C_{i_j})$$

Therefore

$$\rho(C)\delta(C_{i_{j+1}}) - \delta(C)\rho(C_{i_{j+1}}) \leq \rho(C)\delta(C_{i_j}) - \delta(C)\rho(C_{i_j}) \tag{19}$$

Finally, the expression of $\rho(C)\delta(C_{i_j}) - \delta(C)\rho(C_{i_j})$ is in the interval $[\rho(C)m, -\delta(C)m]$ and it is decreased every time when C is found. This proves the lemma. □

In summary, the algorithm can find at most one set with particular values of $\delta(C), \rho(C)$ and it can be found at most $(\rho(C)+\delta(C))m$ times. Then the total number of iterations is at most $\sum_{\delta=0}^{m}\sum_{\rho=0}^{n}(\delta+\rho)m = m^3(m+1)$.

Theorem 17 *Algorithm 2 terminates after at most $\mathcal{O}(m^4)$ iterations.*

The running time of an iteration is dominated by the submodular function minimization, that is every iteration takes $\mathcal{O}(n^5T + n^6)$ time [5].

To sum up, we get that

Theorem 18 *Running time of Algorithm 2 is $\mathcal{O}(m^4T) = \mathcal{O}(m^4n^5T + m^4n^6)$.*

Note, the algorithm above can be improved by using the sets that are found in a previous iteration. With this technique, the algorithm runs at most $\mathcal{O}(m^2)$ iterations. The running time of an iteration is still dominated by the submodular function minimization, that is every iteration takes $\mathcal{O}(n^5T + n^6)$ time [5]. Therefore these algorithms solve the Balanced Submodular Flow problem in $\mathcal{O}(m^2T) = \mathcal{O}(m^2n^5T+m^2n^6)$ time.

4 Balanced integral submodular flows

The simple example of a graph having two nodes and two parallel edges shows that the minimum spread solution is not always possible to be chosen to be integral, even in case of the simple network flows with integer supply vector. This section shows how an integral flow of minimum spread can be found.

From now on, let us assume that b is integral.

Definition 19 *For an arbitrary integer value $\kappa \in \mathbb{Z}$, let $s_I(\kappa)$ denote the minimal value σ for which there exists an integral $(\kappa\mathbb{1}, (\kappa+\sigma)\mathbb{1})$-bounded submodular flow $x \in \mathbb{Z}^A$.*

Claim 20 *For any $\kappa \in \mathbb{Z}$, $s_I(\kappa) = \lceil s(\kappa) \rceil$.*

Proof: Clearly, $s_I(\kappa) \geq s(\kappa)$. On the other hand, the definition of $s(\kappa)$ implies the existence of a $(\kappa\mathbb{1}, (\kappa + s(\kappa))\,\mathbb{1})$-bounded submodular flow x. This flow is also bounded by the integer vectors $\kappa\mathbb{1}$ and $(\kappa + \lceil s(\kappa) \rceil)\,\mathbb{1}$, therefore an integer submodular flow must also exist between these bounds.[4] □

The claim above and the convexity of $s(\kappa)$ immediately gives the following.

Claim 21 $S_I(p) = \min\{s_I(\lfloor \sigma^* \rfloor), s_I(\lceil \sigma^* \rceil)\} = \min\{\lceil s(\lfloor \sigma^* \rfloor) \rceil, \lceil s(\lceil \sigma^* \rceil) \rceil\}$

5 Weighted Balanced Submodular Flows

In this section, the Weighted Balanced Submodular Flows Problem is introduced, and then a natural approach to minimize a single variable convex function is described, that is used for the Weighted Submodular Flow Problem.

Definition 22 *Given an edge weight $c : A \longrightarrow \mathbb{R}^+$, the* weighted spread $\sigma(cx)$ *of a vector $x \in \mathbb{R}^A$ is the value*

$$\max_{a \in A} c(a)x(a) - \min_{a \in A} c(a)x(a)$$

The *Balanced Submodular Flow Problem* is stated as follows.

Problem 23 *For a given directed graph $G = (V, A)$, an edge weight $c : A \longrightarrow \mathbb{R}^+$ and a submodular function $b : \mathcal{P}(V) \longrightarrow \mathbb{Z}$, the aim is to find a submodular flow of minimum weighted spread, i.e find*

$$\sigma^* := \min\left\{\sigma(cx) : x \in \mathbb{R}^A, \varrho_x(X) - \delta_x(X) \leq b(X) \quad \forall X \subseteq V\right\} \tag{20}$$

along with a minimizing vector x.

We will use the following notations:

$$n := |V|, \qquad m := |A|, \qquad \varrho_{\frac{1}{c}}(X) := \sum_{e \in \varrho(X)} \frac{1}{c(e)}$$

$$\delta_{\frac{1}{c}}(X) := \sum_{e \in \delta(X)} \frac{1}{c(e)}, \qquad \partial_{\frac{1}{c}}(X) := \varrho_{\frac{1}{c}}(X) - \delta_{\frac{1}{c}}(X)$$

Definition 24 *For an arbitrary real value $\kappa \in \mathbb{R}$, let $s(\kappa)$ denote the minimal value σ for which there exists a $\left(\kappa\frac{1}{c}, (\kappa+\sigma)\frac{1}{c}\right)$-bounded submodular flow x.*

From Theorem 5 it follows that

$$s(\kappa) = \min\left\{\sigma \geq 0 : \kappa\partial_{\frac{1}{c}}(X) - b(X) \leq \sigma\delta_{\frac{1}{c}}(X) \quad \forall X \subseteq V\right\} \tag{21}$$

5.1 Handler-Zang Method for Convex Function Minimization

In case of piecewise linear functions, it is able to find the exact optimum in finite number of iteration, which makes it a useful tool for solving certain parametric combinatorial optimization problems. Its first use in this scenario is probably due to Handler and Zang[2].

Let $f : \mathbb{R} \longrightarrow \mathbb{R}$ be a convex function, and let us consider Algorithm 3.

Algorithm 3 Handler-Zang method

Let $a_1, b_1 \in R$ such that $a_1 \leq \arg\min f \leq b_1$
Let $\alpha_1 \in \partial f(a_1)$ and $\beta_1 \in \partial f(b_1)$
$i := 1$
loop
    Let $c_i := \frac{f(b_i)-f(a_i)+\alpha_i a_i-\beta_i b_i}{\alpha_i-\beta_i}$
    Let $\sigma_i := \frac{\alpha_i f(b_i)-\beta_i f(a_i)+\alpha_i\beta_i(a_i-b_i)}{\alpha_i-\beta_i}$
    if $f(c_i) = \sigma_i$ **then** RETURN c_i
    end if
    Let γ_i be a subgradient of f at c_i
    if $\gamma_i < 0$ **then**
        $a_{i+1} := c_i$, $\alpha_{i+1} := \gamma_i$,
        $b_{i+1} := b_i$, $\beta_{i+1} := \beta_i$
    else
        $a_{i+1} := a_i$, $\alpha_{i+1} := \alpha_i$,
        $b_{i+1} := c_i$, $\beta_{i+1} := \gamma_i$
    end if
    $i \longleftarrow i + 1$
end loop

Claim 25 *The following statements are true*

- $a_1 \leq a_2 \leq a_3 \leq \cdots$ *and* $b_1 \geq b_2 \geq b_3 \geq \cdots$
- $\alpha_1 \leq \alpha_2 \leq \alpha_3 \leq \cdots$ *and* $\beta_1 \leq \beta_2 \leq \beta_3 \leq \cdots$
- *For each iteration* i *either of the following holds*
    1. $a_i < a_{i+1}$ *and* $\alpha_i < \alpha_{i+1}$
    2. $b_i > b_{i+1}$ *and* $\beta_i > \beta_{i+1}$
- *The subgradients* $\gamma_i \quad (i = 1, 2, \dots)$ *computed during the execution are all different.*

Claim 26 *If* f *is a piecewise linear convex function, then Algorithm 3 finds the minimum of* f *in finite number of steps, and the number of iterations are at most the number of linear segments of* f*.*

In order to be able to start this algorithm, we need to find an initial interval $[a_1, b_1]$ including the optimum. Assuming that $f \geq 0$, this can be done by starting with and arbitrary value and iterating the usual Newton-Dinkelbach steps until either we find a value the with a subgradient of the opposite sign — or one with subgradient equal to 0 meaning that the minimum is already found. The number of these iterations are also limited by the number of linear segments of f.

Note that even when the function has exponentially many linear segment, strongly polynomial bound on the number of iteration can be proven for certain classes convex functions, see [6].

5.2 Handler-Zang method for Weighted Submodular Balanced Flow Problem

Theorem 27 *Assume that there exist a set $X \subseteq V$ such that $\partial_{\frac{1}{c}}(X) > 0$ then*

$$\sigma^* = \max\left\{ \frac{-b(X)\partial_{\frac{1}{c}}(Y) + b(Y)\partial_{\frac{1}{c}}(X)}{\delta_{\frac{1}{c}}(X)\partial_{\frac{1}{c}}(Y) - \delta_{\frac{1}{c}}(Y)\partial_{\frac{1}{c}}(X)} \mid \partial_{\frac{1}{c}}(X) \geq 0, \partial_{\frac{1}{c}}(Y) < 0 \right\}$$

and σ^* along with the maximizing sets $X, Y \subseteq V$ and it can be computed with $\mathcal{O}(n^4 m^8 \log^6(m))$ submodular function minimization problems.

Note that in case $\partial_{\frac{1}{c}}(X) = 0$ holds for all $X \subseteq V$, $s(\kappa)$ is a constant function. We get that

$$\sigma^* = \max_{\partial_{\frac{1}{c}}(X)=0} \left\{ \frac{-b(X)}{\delta_{\frac{1}{c}}(C)} \mid b(X) < 0 \right\}$$

and σ^* along with the maximizing set $X \subseteq V$ can be computed is $\mathcal{O}(m^2)$ submodular function minimization problems. If there exist $X \subseteq V$ such that $b(X) < 0$ and $\delta_{\frac{1}{c}}(X) = 0$, the problem must be infeasible.

The expression in the above theorem is a lower bound for σ^* can be proven in a similar as in the non-weighted case. Furthermore, the maximizing sets X, Y can be chosen to satisfy $X \subset Y$ or $Y \subset X$. To prove the theorem, it is sufficient to give an algorithm for finding X^*, Y^* for which the equality holds.

By definition of $s(\kappa)$ and claim 10, σ^* is equal to the minimum of $s(\kappa)$. Handler-Zang method can be used for finding this minimum. First, an initial interval is needed that contains the optimum κ^*. Due to Theorem 5

$$\kappa^* \leq \frac{b(X)}{\varrho_{\frac{1}{c}}(X)}$$

for all $X \subseteq V$ such that $\delta_{\frac{1}{c}}(X) = 0$ and $\varrho_{\frac{1}{c}}(X) > 0$. Let us consider the following minimum:

$$\kappa' = \min\left\{ \frac{b(X)}{\varrho_{\frac{1}{c}}(X)} \mid \delta_{\frac{1}{c}}(X) = 0, \varrho_{\frac{1}{c}}(X) > 0 \right\}$$

It can be computed with $\mathcal{O}(m^2)$ submodular function minimization, as a consequence of the next theorem proved by M. X. Goemans at all. [11]:

Theorem 28 *Let b be a submodular function on V and $a(S) = \sum_{s \in S} a_s$ a linear function, where $|V| = n$ and $a \in \mathbf{R^n}$. We define δ^* as follows:*

$$\delta^* = \max\left\{ \delta \mid \min_{S \subset V} b(S) - \delta a(S) \geq 0 \right\} = \min_{S \subset V}\left\{ \frac{b(S)}{a(S)} \mid a(S) > 0 \right\} \tag{22}$$

Then δ^* can be computed using the discrete Newton's algorithm and it takes $\mathcal{O}(n^2)$ iterations.

The initial interval can be found by starting with κ' and applying the same technique as with the Handler-Zang method. Since the considered κ is less than κ', $s(\kappa)$ is unrelated to sets such that $\delta_{\frac{1}{c}}(X) = 0$. At this point, 21 can be rewritten as follows:

$$s(\kappa) = \max\left\{ \frac{\kappa\partial_{\frac{1}{c}}(X) - b(X)}{\delta_{\frac{1}{c}}(X)} \mid \delta_{\frac{1}{c}}(X) > 0 \right\}$$

In other words, $s(\kappa)$ is a maximum of the linear functions

$$\kappa\frac{\partial_{\frac{1}{c}}(X)}{\delta_{\frac{1}{c}}(X)} - \frac{b(X)}{\delta_{\frac{1}{c}}(X)}$$

Thus, it is a piecewise linear function and for any κ and $\frac{\partial_{\frac{1}{c}}(X)}{\delta_{\frac{1}{c}}(X)}$ is a subgradient of $s(\kappa)$. Since $b(X) - \kappa\partial_{\frac{1}{c}}(X)$ is submodular and $\delta_{\frac{1}{c}}(X)$ is linear, $s(\kappa)$ can be computed with the discrete Newton's method with $\mathcal{O}(^2)$ iterations by Theorem 28.

To prove the equation in Theorem 27, the minimum of $s(\kappa)$ is required and the Handler-Zang method can be used. Let κ^* be the point at which the value of $s(\kappa)$ is minimal. Let κ^* be the point at which the value of $s(\kappa)$ is minimal. Let us define $\overline{s}(\kappa) = s(\kappa) - s(\kappa^*)$. Then the number of iterations required to find the minimum point of $s(\kappa)$ is less than twice the iteration number to find the root of $\overline{s}(\kappa)$. Hence, it's enough to estimate the number of steps to find this root. The following theorem can be proven:

Theorem 29 *The minimum of $s(\kappa)$ can be computed using the Handler-Zang method with at most $\mathcal{O}(n^4m^6\log^6(m))$ iterations. The maximizing sets X, Y in Theorem 27 are the endpoints of the interval in the last iteration.*

Acknowledgement

The authors would like to acknowledge the valuable suggestions of András Frank.

References

[1] M. G. Scutellà, A Strongly Polynomial Algorithm for the Uniform Balanced Network Flow Problem, *Discret. Appl. Math.* **81** (1998)

[2] Handler, Gabriel Y. and Zang, Israel, A dual algorithm for the constrained shortest path problem, *Networks* **10** (1980)

[3] András Frank, Finding feasible vectors of Edmonds-Giles polyhedra, *Journal of Combinatorial Theory, Series B* **36** (1984)

[4] P.L. Hammer and E.L. Johnson and B.H. Korte and G.L. Nemhauser, A Min-Max Relation for Submodular Functions on Graphs, *Annals of Discrete Mathematics* **1** (1977)

[5] Orlin, James, A Faster Strongly Polynomial Time Algorithm for Submodular Function Minimization, *Mathematical Programming* **118** (2007)

[6] Alpár Jüttner, On Resource Constrained Optimization Problems, *4th Japanese-Hungarian Symposium on Discrete Mathematics and Its Applications* (2005)

[7] S Martello and W.R Pulleyblank and P Toth and D de Werra, Balanced optimization problems, *Operations Research Letters* **3** (1984)

[8] Ravindra K. Ahuja, The balanced linear programming problem, *European Journal of Operational Research* **101** (1997)

[9] Wu, Longshu, An Efficient Algorithm for the Most Balanced Spanning Tree Problems, *Advanced Science Letters* **11** (2012)

[10] Klinz Bettina, Maria Grazia Scutellà, A Strongly Polynomial Algorithm for the Balanced Network Flow Problem, (2000)

[11] Goemans, Michel X. and Gupta, Swati and Jaillet, Patrick, Discrete Newton's Algorithm for Parametric Submodular Function Minimization, *Integer Programming and Combinatorial Optimization* (2017)

[12] Tomasz Radzik, Fractional combinatorial optimization", *Handbook of Combinatorial Optimization* (1998)

The GRAPH of graphs of optimal subsets of pairwise comparisons

Zsombor Szádoczki

Research Laboratory on Engineering & Management Intelligence
Institute for Computer Science and Control (SZTAKI),
Eötvös Loránd Research Network (ELKH)
1111 Kende u. 13-17., Budapest, Hungary;
Department of Operations Research and Actuarial Sciences
Corvinus University of Budapest
1093 Fővám tér 8., Budapest, Hungary
szadoczki.zsombor@sztaki.hu

Sándor Bozóki

Research Laboratory on Engineering & Management Intelligence
Institute for Computer Science and Control (SZTAKI),
Eötvös Loránd Research Network (ELKH)
1111 Kende u. 13-17., Budapest, Hungary;
Department of Operations Research and Actuarial Sciences
Corvinus University of Budapest
1093 Fővám tér 8., Budapest, Hungary
bozoki.sandor@sztaki.hu

Abstract: Pairwise comparisons form the corner stone of ranking, preference modelling and multi-attribute decision making. We are focusing on incomplete pairwise comparison matrices using their graph representation. In this paper the optimal subsets of comparisons – i.e., the ones that provide the closest logarithmic least squares weight vectors to the vectors calculated from the complete case – are determined for the given numbers of items to compare and comparisons. Simulations are used to find the optimal subsets, which result in a GRAPH of graphs for a given number of alternatives. Regularity and bipartiteness are the most important properties of the optimal graphs, which can often be reached from each other by adding (deleting) exactly one comparison. The sequences of comparisons gained this way can be particularly useful in those problems, when the number of comparisons provided by the decision maker is uncertain (e.g., online questionnaires).

Keywords: multi-attribute decision making, pairwise comparison, incomplete pairwise comparison matrix, representing graph, GRAPH of graphs

1 Introduction

Pairwise comparisons are fundamental in ranking, preference modelling, multi-attribute decision making (MADM), and even in sports. We focus on the incomplete case of pairwise comparison matrices (PCMs), which are frequently used in the popular decision making methodology, the Analytic Hierarchy Process (AHP) [7]. Incompleteness means that some comparisons are missing. We can still determine a prioritization vector (if some general condition hold) of the alternatives in this case, however, the number of known comparisons (e) and their arrangement has a crucial effect on the outcomes. The subset of the known comparisons for a given n number of alternatives is often represented by graphs [1], where the vertices correspond to the items to be compared, and there is an edge between two vertices, if the appropriate comparison is known.

In this study, we determine the optimal subsets of pairwise comparisons, namely, the ones that on average provide the closest logarithmic least squares weight vectors to the ones computed from the complete case for a given number of alternatives (n) and a given number of comparisons (e). We also rely on the graph representation of the pairwise comparisons, and apply a GRAPH of graphs to visualize our findings. VERTICEs of a GRAPH are graphs, and there is an EDGE between two VERTICEs (=graphs)

if the associated graphs are in a specified relation, e.g., as in our case, they can be drawn from each other by adding or deleting exactly one edge.

Depending on the specification of the relation, several GRAPHs of graphs have been investigated. Bondy and Lovász (see [5, Theorem 2]) showed that the GRAPH of graphs is connected, where GRAPH is defined as follows: let G be a 2-connected graph on n nodes, v is a node of G; NODEs are the spanning trees of G, and two NODEs are connected by an EDGE if the corresponding spanning trees have a common subtree on $n-1$ nodes including v.

Another remarkable GRAPH of graphs is the Petersen family of seven graphs, including the Petersen graph itself. Two graphs are connected by an EDGE if they can be transformed from each other by replacing a triangle by a 3-star (including the addition of its center), see e.g. [3, page 2].

The GRAPH of graphs by [6] is motivated by the evolution of graphs in a dynamic system.

It is worth noting that the term 'neighbouring graphs' in [5] is used synonymously for 'there is an EDGE between two graphs'. Analogously, 'reachable' in [6] means that there is a PATH between two graphs. We use the concept of GRAPH of graphs to visualize our results throughout the paper.

The optimal subsets for the investigated (n, e) pairs are important results on their own, but in addition, using that some of them are reachable from each other, by adding (deleting) exactly one comparison, one can also create optimal sequences of comparisons. This can be crucial in order to estimate the decision makers' preferences the best way, when the number of comparisons provided is a priori uncertain (for instance in online questionnaires).

2 Main results

We apply extensive numerical simulations with a sample size of 1 million random pairwise comparison matrices to compare all the possible different subsets of comparisons for given pairs of (n, e). The used simulation approach has been applied for some special cases in [8]. The computed weight vectors are compared to the vectors gained from the complete matrix (complete set of comparisons) with the Kendall's rank correlation coefficient and the Euclidean distance measures. We mainly focus on the logarithmic least squares weight calculation technique, when a logarithmic least squares objective function is minimized based on the known comparisons. Some calculations have also been carried out for the eigenvector method, but the optimal graphs were always identical.

If the system of comparisons does not contain any contradiction, then every weight calculation technique provides the same proiritization vector. However, when a decision maker compare the possible alternatives, it often occurs that A is 2 times as good as B, while B is 3 times as good as C, but A is not 6 times as good as C. There is an inconsistency in the system that affects the calculated weights. In our simulations we use three different inconsistency levels to account for this.

It turns out that the different metrics (Kendall's tau or Euclidean distance) and inconsistency levels provide the same results, namely the best subset of comparisons for a given (n, e) pair is practically always the same. This way the optimal graphs (subsets) are determined for all the possible (n, e) pairs in case of $n \leq 6$. It is also important to note that [8] used several different metrics on the special cases examined by them – e.g. the dice [10], and cosine similarity [4] measures – and found that the results are not depending on the chosen measures. The findings of [9] also suggest that even the standard deviations of the different metrics provide the same ranking of the competing patterns (graphs). Moreover, [2] find that our results are also relevant outside of the domain of pairwise comparison matrices, as exactly the same graphs are the optimal ones for every (n, e) pair examined here in the case of other paired comparison-based models, e.g., the Bradley-Terry and the Thurstone models.

Many of the found optimal subsets are reachable from each other resulting in optimal sequences of comparisons that can be presented in a GRAPH of graphs.

For the sake of brevity, we only include a smaller portion of the optimal graphs with different parameters. The GRAPH of graphs for $n = 5$ can be seen in Figure 1 focusing on the connected cases. The optimal graphs as well as the optimal sequences are highlighted by green color.

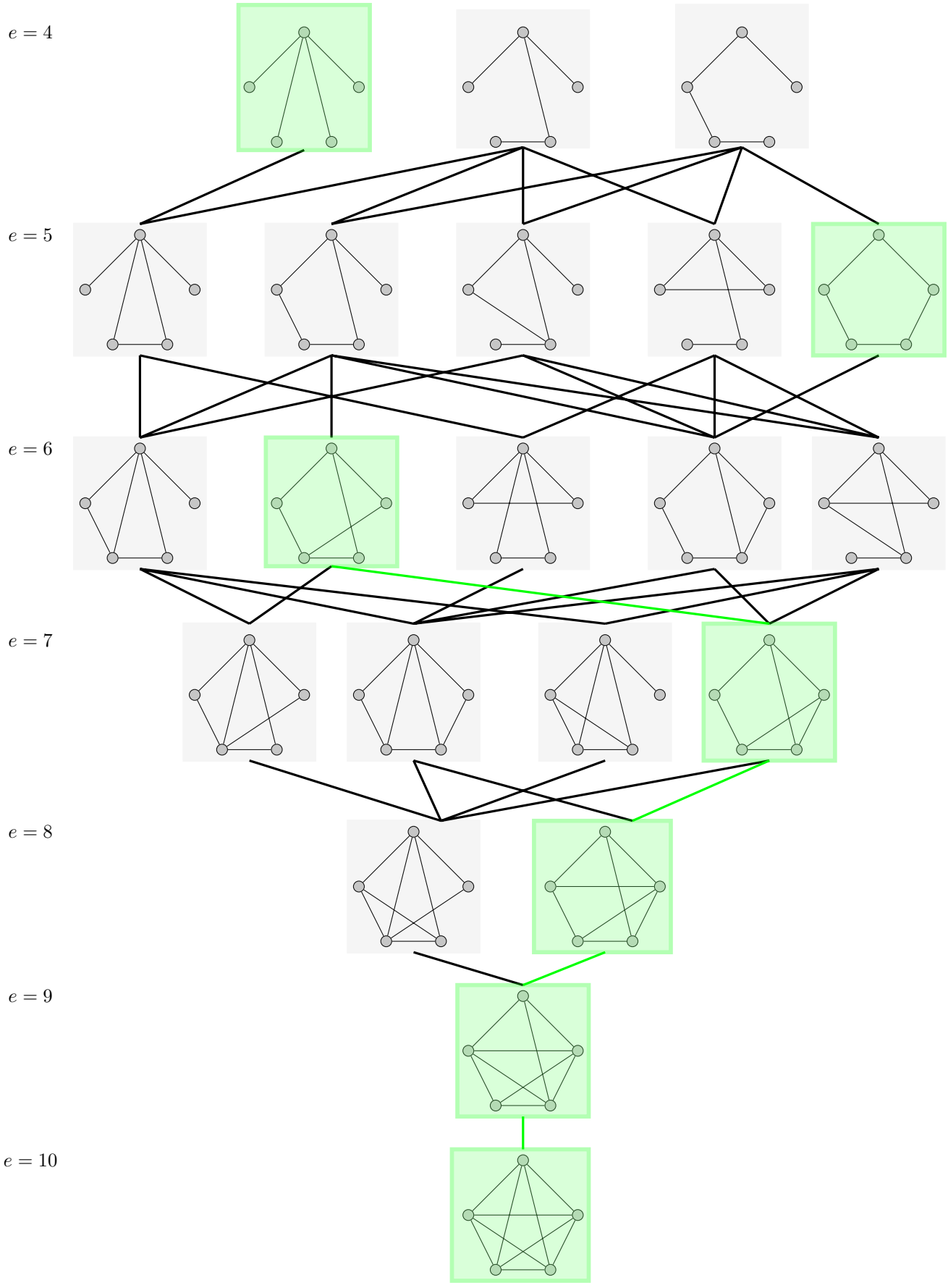


Figure 1: The GRAPH of graphs for $n = 5$, optimal graphs and sequences are highlighted by green

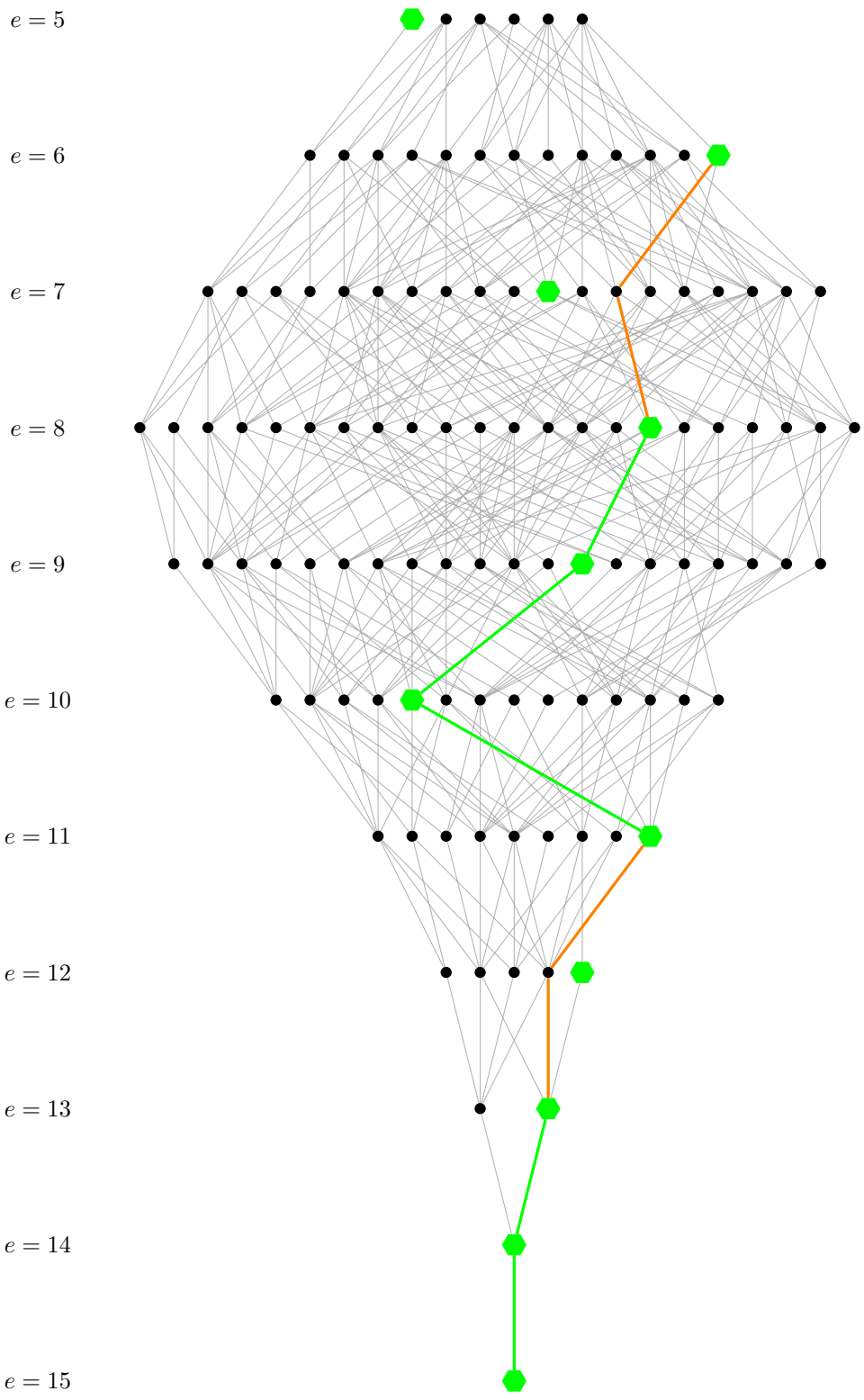


Figure 2: The GRAPH of graphs for $n = 6$

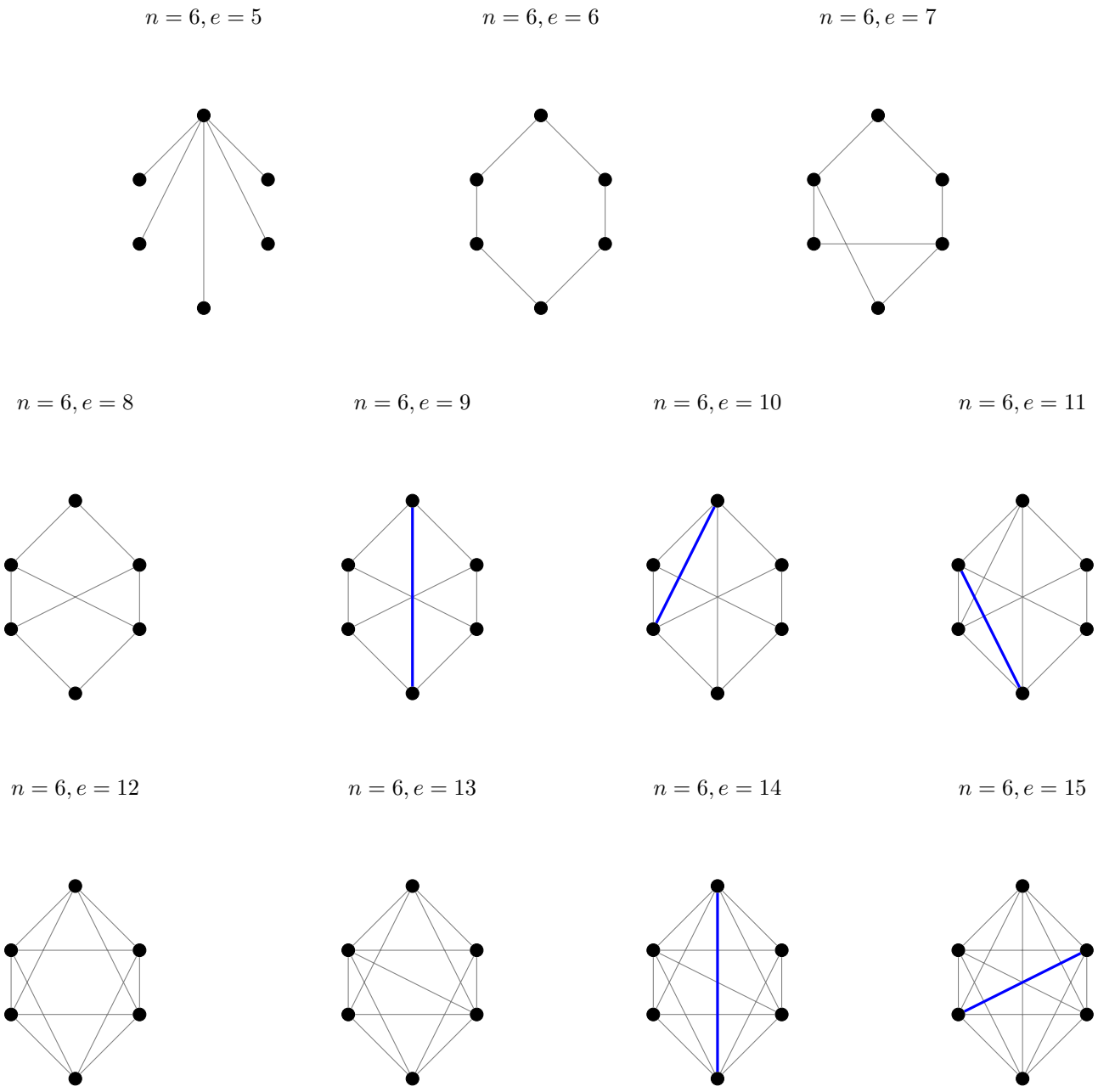


Figure 3: The optimal graphs for $n = 6$, corresponding to the green hexagonal NODES in Figure 2

The GRAPH of graphs for $n = 6$ containing all the possible subsets of pairwise comparisons in the case of connected representing graphs can be seen in Figure 2. To keep the visibility of the figure, the small graphs are only represented by nodes, while the optimal cases are highlighted by green color, and they are also detailed in Figure 3, where the newly added comparison is highlighted by blue color in every step when possible.

As one can see, all the optimal graphs are generally not reachable from each other, however, one can

determine a sequence of comparisons that contain as many optimal graphs as possible, and the remaining cases are also close to the optimal ones. These latter comparisons are highlighted by orange in Figure 2. Interestingly, k-regular graphs are always optimal for $(n, e = k \cdot n/2)$ parameters, but regularity is a key property in general: the degree of vertices is always as close to each other as possible. On the other hand, the optimal representing graphs are also always close to bipartite ones. It is worth noting that star graphs are always optimal among spanning trees.

Our findings are instantly applicable in real problems, and an optimal sequence of comparisons can guarantee that we estimate the prioritization vector of decision makers in an optimal way whenever they stop answering the questions determining the comparisons. Moreover, the results seem to be more general considering different kinds of measures and models based on pairwise comparisons [8, 2].

3 Conclusion and future research

We used numerical simulations to determine the optimal subsets of pairwise comparisons for a given number of items to be compared and comparisons. Some of the optimal subsets are reachable from each other by adding (deleting) exactly one comparison. The most important properties of the optimal graphs are regularity and bipartiteness, while star graphs are always optimal among spanning trees. The results are visualized with the help of GRAPHs of graphs, and they can be applied in real problems instantly without any difficulty. The findings can be especially useful in case of online questionnaires, when the decision makers tend to abandon the problem, and the number of provided comparisons is uncertain.

Here we focused on incomplete pairwise comparison matrices, however, the optimality of the found subsets seem to be more general, it applies in other models, such as the Bradley-Terry and the Thurstone models as well [2].

The indirect connections between the different graphs, namely, whether we can reach a graph from another one by adding a given number of comparisons in one step can be an interesting research direction, as well as accounting for a priori information, i.e., labeling the different vertices.

References

[1] Gass, S.I., Tournaments, transitivity and pairwise comparison matrices, *Journal of the Operational Research Society,* **49(6):616–624.** (1998)

[2] Gyarmati, L., Orbán-Mihálykó, É., Mihálykó, Cs., Bozóki, S., and Szádoczki, Zs., The incomplete Analytic Hierarchy Process and Bradley-Terry model: (in)consistency and information retrieval, *arXiv: https://doi.org/10.48550/arxiv.2210.03700,* (2022)

[3] Hashimoto, H. and Nikkuni, R., On Conway–Gordon type theorems for graphs in the Petersen family, *Journal of Knot Theory and Its Ramifications,* **22(9):1350048.** (2013)

[4] Kou, G. and Lin, C., A cosine maximization method for the priority vector derivation in AHP, *European Journal of Operational Research,* **235(1):225–232.** (2014)

[5] Lovász, L., A homology theory for spanning trees of a graph, *Acta Mathematica Academiae Scientiarum Hungaricae,* **30(3-4):241-251.** (1977)

[6] Mesbahi, M., On a dynamic extension of the theory of graphs, *Proceedings of the 2002 American Control Conference (IEEE Cat. No.CH37301),* **2:1234-1239.** (2002)

[7] Saaty, T.L., A scaling method for priorities in hierarchical structures, *Journal of Mathematical Psychology,* **15(3):234-281.** (1977)

[8] Szádoczki, Zs., Bozóki, S., Juhász, P., Kadenko, S. V., and Tsyganok, V., Incomplete pairwise comparison matrices based on graphs with average degree approximately 3, *Annals of Operations Research,* (2022)

[9] Szádoczki, Zs., Bozóki, S., and Tekile, H. A., Filling in pattern designs for incomplete pairwise comparison matrices: (Quasi-)regular graphs with minimal diameter, *Omega,* **107(C):102557.** (2022)

[10] Ye, J., Multicriteria decision-making method using the dice similarity measure based on the reduct intuitionistic fuzzy sets of interval-valued intuitionistic fuzzy sets, *Applied Mathematical Modelling,* **36(9):4466–4472.**(2012)

Packing mixed hyperarborescences

Zoltán Szigeti[1]

Laboratory G-SCOP, University Grenoble Alpes
Grenoble, France
zoltan.szigeti@grenoble-inp.fr

Abstract: The aim of this paper is twofold. We first provide a new orientation theorem which gives a natural and simple proof of a result of Gao, Yang [11] on matroid-reachability-based packing of mixed arborescences in mixed graphs by reducing it to the corresponding theorem of Cs. Király [16] on directed graphs. Moreover, we extend another result of Gao, Yang [12] by providing a new theorem on mixed hypergraphs having a packing of mixed hyperarborescences such that their number is at least ℓ and at most ℓ', each vertex belongs to exactly k of them, and each vertex v is the root of least $f(v)$ and at most $g(v)$ of them.

Keywords: arborescence, mixed hypergraph, packing

1 Introduction

This paper is not a survey on packing arborescences. Such a survey is in preparation, see [19]. We only present here those theorems of the topic that are closely related to the new results of this paper.

Edmonds [5] characterized digraphs having a packing of spanning arborescences with fixed roots. Frank [7] extended it for a packing of spanning arborescences whose roots are not fixed. The result of Frank [7], and independently Cai [3], answers the question when a digraph has an (f, g)-bounded packing of spanning arborescences, that is when each vertex v can be the root of at least $f(v)$ and at most $g(v)$ arborescences in the packing. Bérczi-Frank [2] entends it for an (f, g)-bounded, k-regular, (ℓ, ℓ')-limited packing of not necessarily spanning arborescences, where k-regular means that each vertex belongs to exactly k arborescences in the packing and (ℓ, ℓ')-limited means that the number of arborescences in the packing is at least ℓ and at most ℓ'. Kamiyama, Katoh, Takizawa [15] provided a different type of generalization of Edmonds' theorem in which they wanted to pack reachability arborescences, that is each arborescence in the packing must contain all the vertices that can be reached from its root in the digraph. Durand de Gevigney, Nguyen, Szigeti [4] gave a generalization of Edmonds theorem where a matroid constraint was added for the packing. More precisely, given a matroid M on a multiset of vertices of a digraph D, they wanted to have a matroid-based packing of arborescences, that is for every vertex v of D, the set of roots of the arborescences in the packing containing v must form a basis of M. Cs. Király [16] proposed a common generalization of the previous two results. He characterized pairs (D, M) of a digraph and a matroid that have a matroid-reachability-based packing of arborescences, that is for every vertex v of D, the set of roots of the arborescences in the packing containing v must form a basis of the subset of the elements of M from which v is reachably in D.

All of these results hold for dypergraphs, see [10], [13], [19], [1], [6], and for mixed graphs, see [7], [11], [19], [18], [6], [12]. In fact, all of these results, except the one of Bérczi-Frank [2], are known to hold for mixed hypergraphs, see [6], [13], [14]. The present paper will fill in this gap by showing that this result also holds for mixed hypergraphs. More precisely, we will characterize mixed hypergraphs having an (f, g)-bounded, k-regular, (ℓ, ℓ')-limited packing of mixed hyperarborescences. Our result naturally generalizes a result of Gao, Yang [12] on (f, g)-bounded packing of k spanning mixed arborescences. The other aim of this paper is to provide a new proof of another result of Gao, Yang [11] on matroid-reachability-based packing of mixed arborescences. Our approach is to reduce the result to the result of Cs. Király [16] on matroid-reachability-based packing of arborescences via a new orientation theorem.

2 Definitions

A *multiset* of V may contain multiple occurrences of elements. For a multiset S of V and a subset X of V, $\boldsymbol{S_X}$ denotes the multiset consisting of the elements of X with the same multiplicities as in S.

Let $\boldsymbol{D} = (V, A)$ be a directed graph, shortly *digraph*. For a subset X of V, the set of arcs in A *entering* X is denoted by $\boldsymbol{\rho_A(X)}$ and the *in-degree* of X is $\boldsymbol{d^-_A(X)} = |\rho_A(X)|$. For a subset X of V, we denote by $\boldsymbol{P^X_D}$ ($\boldsymbol{Q^X_D}$) the set of vertices from (to) which there exists a path to (from, respectively) at least one vertex of X. We say that D is an *arborescence with root* s, shortly *s-arborescence*, if $s \in V$ and there exists a unique path from s to v for every $v \in V$; or equivalently, if D contains no circuit and every vertex in $V - s$ has in-degree 1. A subgraph of D is called a *spanning (resp. reachability) s-arborescence* if it is an s-arborescence and its vertex set is V (resp. Q^s_D). By a *packing* of subgraphs in D, we mean a set of subgraphs that are arc-disjoint. A packing of subgraphs is called *k-regular* if every vertex belongs to exactly k subgraphs in the packing. For two functions $f, g : V \to \mathbb{Z}_+$, a packing of arborescences is called *(f, g)-bounded* if the number of v-arborescences in the packing is at least $f(v)$ and at most $g(v)$ for every $v \in V$. For $\ell, \ell' \in \mathbb{Z}_+$, a packing of arborescences is called *(ℓ, ℓ')-limited* if the number of arborescences in the packing is at least ℓ and at most ℓ'. For a multiset S of V and a matroid M on S, a packing of arborescences in D is called *matroid-based* (resp. *matroid-reachability-based*) if every $s \in S$ is the root of at most one arborescence in the packing and for every $v \in V$, the multiset of roots of arborescences containing v in the packing forms a basis of S (resp. $S_{P^v_D}$) in M.

Let $\boldsymbol{F} = (V, E \cup A)$ be a *mixed graph*, where E is a set of edges and A is a set of arcs. A mixed subgraph F' of F is a *mixed path* if the edges in F' can be oriented in such a way that we obtain a directed path. For a subset X of V, we denote by $\boldsymbol{P^X_F}$ ($\boldsymbol{Q^X_F}$) the set of vertices from (to) which there exists a mixed path to (from, respectively) at least one vertex of X. We say that F is *strongly connected* if there exists a mixed path from s to t for all $(s, t) \in V^2$. The maximal strongly connected subgraphs of F are called the *strongly connected components* of F. A *mixed s-arborescence* is a mixed graph that has an orientation that is an s-arborescence. A mixed subgraph of F is called a *spanning (resp. reachability) mixed s-arborescence* if it is a mixed s-arborescence and its vertex set is V (resp. Q^s_F). By a *packing* of subgraphs in F, we mean a set of subgraphs that are edge- and arc-disjoint. All the packing problems considered in digraphs can also be considered in mixed graphs.

Let $\boldsymbol{\mathcal{D}} = (V, \mathcal{A})$ be a directed hypergraph, shortly *dypergraph*, where $\boldsymbol{\mathcal{A}}$ is the set of dyperedges of $\mathcal{D}$. A *dyperedge* e is an ordered pair (Z, z), where $z \in V$ is the *head* and $\emptyset \neq Z \subseteq V - z$ is the set of *tails* of e. For $X \subseteq V$, a dyperedge (Z, z) *enters* X if $z \in X$ and $Z \cap \overline{X} \neq \emptyset$. The set of dyperedges in $\mathcal{A}$ *entering* X is denoted by $\boldsymbol{\rho_{\mathcal{A}}(X)}$ and the *in-degree* of X is $\boldsymbol{d^-_{\mathcal{A}}(X)} = |\rho_{\mathcal{A}}(X)|$. By *trimming* a dyperedge $e = (Z, z)$, we mean the operation that replaces e by an arc yz where $y \in Z$. We say that $\mathcal{D}$ is a *hyperarborescence with root* s, shortly *s-hyperarborescence*, if $\mathcal{D}$ can be trimmed to an s-arborescence. A *packing* of subdypergraphs in $\mathcal{D}$ is a set of subdypergraphs that are dyperedge-disjoint. We say that $\mathcal{D}$ has a *matroid-based/(f, g)-bounded/k-regular/(ℓ, ℓ')-limited* packing of hyperarborescences if $\mathcal{D}$ can be trimmed to a digraph that has a matroid-based/(f, g)-bounded/k-regular/(ℓ, ℓ')-limited packing of arborescences.

Let $\boldsymbol{\mathcal{F}} = (V, \mathcal{E} \cup \mathcal{A})$ be a *mixed hypergraph*, where $\boldsymbol{\mathcal{E}}$ is the set of hyperedges and $\boldsymbol{\mathcal{A}}$ is the set of dyperedges of $\mathcal{F}$. A *hyperedge* is a subset of V of size at least two. A hyperedge e *enters* a subset Y of V if $e \cap Y \neq \emptyset \neq \overline{e} \cap Y$. By *orienting* a hyperedge e, we mean the operation that replaces the hyperedge e by a dyperedge $(e - x, x)$ for some $x \in e$. For $\vec{\mathcal{Z}} \subseteq \mathcal{A}$, $\mathcal{Z}$ denotes the set of underlying hyperedges of $\vec{\mathcal{Z}}$. For $\mathcal{Z} \subseteq \mathcal{E}$ and $X \subseteq V$, we denote by $\boldsymbol{V(\mathcal{Z})}$ the set of vertices that belong to at least one hyperedge in $\mathcal{Z}$ and by $\boldsymbol{\mathcal{Z}(X)}$ the set of hyperedges in $\mathcal{Z}$ that are contained in X. A *mixed s-hyperarborescence* is a mixed hypergraph that has an orientation that is an s-hyperarborescence. A mixed s-hyperarborescence is called *spanning* in $\mathcal{F}$ if its vertex set is V. For a family $\mathcal{P}$ of subsets of V, we denote by $\boldsymbol{e_{\mathcal{E}\cup\mathcal{A}}(\mathcal{P})}$ the number of hyperedges in $\mathcal{E}$ and dyperedges in $\mathcal{A}$ that enter some member of $\mathcal{P}$. For $X \subseteq V$, we use $e_{\mathcal{E}\cup\mathcal{A}}(X)$ for $e_{\mathcal{E}\cup\mathcal{A}}(\{X\})$. A *packing* of mixed subhypergraphs in $\mathcal{F}$ is a set of mixed subhypergraphs that are hyperedge- and dyperedge-disjoint. We say that $\mathcal{F}$ has an *(f, g)-bounded/k-regular/(ℓ, ℓ')-limited* packing of mixed hyperarborescences if $\mathcal{E}$ has an orientation $\vec{\mathcal{E}}$ such that the dypergraph $(V, \vec{\mathcal{E}} \cup \mathcal{A})$ has an (f, g)-bounded/k-regular/(ℓ, ℓ')-limited packing of hyperarborescences.

3 Known results

In this section we list the results on packing arborescences that are related to the new results. We start with the fundamental result of Edmonds [5] on packing spanning arborescences with fixed roots.

Theorem 1 (Edmonds [5]) *Let $D = (V, A)$ be a digraph and S a multiset of V. There exists a packing of spanning s-arborescences ($s \in S$) in D if and only if $d_A^-(X) \geq |S_{V-X}|$ for every $\emptyset \neq X \subseteq V$.*

Theorem 1 was extended for the case when the roots of the arborescences are not fixed but the number of arborescences in the packing rooted at any vertex is bounded. For a subpartition $\mathcal{P}$ of V, $\cup\boldsymbol{\mathcal{P}}$ denotes the set of elements of V that belong to some member of $\mathcal{P}$.

Theorem 2 (Frank [7], Cai [3]) *Let $D = (V, A)$ be a digraph, $f, g : V \to \mathbb{Z}_+$ functions and $k \in \mathbb{Z}_+$. There exists an (f, g)-bounded packing of k spanning arborescences in D if and only if*

$$\begin{aligned} g(v) &\geq f(v) && \text{for every } v \in V, && (1)\\ e_A(\mathcal{P}) &\geq k|\mathcal{P}| - \min\{k - f(\overline{\cup\mathcal{P}}), g(\cup\mathcal{P})\} && \text{for every subpartition } \mathcal{P} \text{ of } V. && (2) \end{aligned}$$

If $f(v) = g(v) = |S_v|$ for every $v \in V - s$, then Theorem 2 reduces to Theorem 1.

Theorem 2 can be generalized for the case when the arborescences are not necessarily spanning but every vertex must belong to the same number of arborescences in the packing. For $g : V \to \mathbb{Z}_+$ and $k \in \mathbb{Z}_+$, let $\boldsymbol{g_k}(v) = \min\{k, g(v)\}$ for every $v \in V$. For convenience, we present not the original version of the result of [2] which is about packing branchings but one that fits better to our framework.

Theorem 3 (Bérczi-Frank [2]) *Let $D = (V, A)$ be a digraph, $f, g : V \to \mathbb{Z}_+$ functions and $k, \ell, \ell' \in \mathbb{Z}_+$. There exists an (f, g)-bounded k-regular (ℓ, ℓ')-limited packing of arborescences in D if and only if*

$$\begin{aligned} g_k(v) &\geq f(v) && \text{for every } v \in V, && (3)\\ \min\{g_k(V), \ell'\} &\geq \ell && && (4)\\ e_A(\mathcal{P}) &\geq k|\mathcal{P}| - \min\{\ell' - f(\overline{\cup\mathcal{P}}), g(\cup\mathcal{P})\} && \text{for every subpartition } \mathcal{P} \text{ of } V. && (5) \end{aligned}$$

For $k = \ell = \ell'$, Theorem 3 reduces to Theorem 2.

An elegant extention of Theorem 1 for packing reachability arborescences was provided in [15].

Theorem 4 (Kamiyama, Katoh, Takizawa [15]) *Let $D = (V, A)$ be a digraph and S a multiset of V. There exists a packing of reachability s-arborescences ($s \in S$) in D if and only if $d_A^-(X) \geq |S_{P_D^X - X}|$ for every $X \subseteq V$.*

When each vertex is reachable from every vertex of S, Theorem 4 reduces to Theorem 1. Theorem 4 can be proved by induction and using Edmonds' result on packing branchings, see Hörsch-Szigeti [14].

Another type of generalizations of Theorem 1 was obtained by adding a matroid constraint.

Theorem 5 (Durand de Gevigney, Nguyen, Szigeti [4]) *Let $D = (V, A)$ be a digraph, S a multiset of V and $\mathsf{M} = (S, r_{\mathsf{M}})$ a matroid. There exists a M-based packing of arborescences in D if and only if $r_{\mathsf{M}}(S_X) + d_A^-(X) \geq r_{\mathsf{M}}(S)$ for every $\emptyset \neq X \subseteq V$.*

For the free matroid M, Theorem 5 reduces to Theorem 1.

A common generalization of Theorems 4 and 5 was found by Cs. Király [16].

Theorem 6 (Cs. Király [16]) *Let $D = (V, A)$ be a digraph, S a multiset of V and $\mathsf{M} = (S, r_{\mathsf{M}})$ a matroid. There exists a matroid-reachability-based packing of arborescences in D if and only if*

$$r_{\mathsf{M}}(S_X) + d_A^-(X) \geq r_{\mathsf{M}}(S_{P_D^X}) \quad \text{for every } X \subseteq V. \qquad (6)$$

For the free matroid M, Theorem 6 reduces to Theorem 4. When each vertex is reachable from a basis of M, Theorem 6 reduces to Theorem 5.

Gao, Yang [11] provided another characterization of the existence of a matroid-reachability-based packing of arborescences.

Theorem 7 (Gao, Yang [11]) *Let $D = (V, A)$ be a digraph, S a multiset of V and $\mathsf{M} = (S, r_{\mathsf{M}})$ a matroid. There exists a matroid-reachability-based packing of arborescences in D if and only if for every strongly connected component C of D and every $X \subseteq P_D^C$ such that $X \cap C \neq \emptyset$ and $d_A^-(X - C) = 0$, $d_A^-(X) \geq r_{\mathsf{M}}(S_{P_D^C}) - r_{\mathsf{M}}(S_X)$.*

Theorems 6 and 7 are equivalent. One implication is shown in [11], the other one in [19].

For dypergraphs we present a generalization of Theorem 5 that will be applied in the proof of the main result of this paper.

Theorem 8 (Fortier, Cs. Király, Léonard, Szigeti, Talon [6]) *Let $\mathcal{D} = (V, \mathcal{A})$ be a dypergraph, S a multiset of V and $\mathsf{M} = (S, \mathcal{I}_{\mathsf{M}})$ a matroid with rank function r_{M}. There exists a M-based packing of hyperarborescences in $\mathcal{D}$ if and only if*

$$r_{\mathsf{M}}(S_X) + d_{\mathcal{A}}^-(X) \quad \geq \quad r_{\mathsf{M}}(S) \qquad \text{for every } \emptyset \neq X \subseteq V. \tag{7}$$

Furthermore, if we want S to be the root set of the arborescences in the packing then (8) must also hold

$$S_v \quad \in \quad \mathcal{I}_{\mathsf{M}} \qquad \text{for every } v \in V. \tag{8}$$

Theorem 2 was generalized for mixed graphs as follows.

Theorem 9 (Gao, Yang [12]) *Let $F = (V, E \cup A)$ be a mixed graph, $f, g : V \to \mathbb{Z}_+$ functions, and $k \in \mathbb{Z}_+$. There exists an (f, g)-bounded packing of k spanning mixed arborescences in F if and only if (1) holds and $e_{E \cup A}(\mathcal{P}) \geq k|\mathcal{P}| - \min\{k - f(\overline{\cup\mathcal{P}}), g(\cup\mathcal{P})\}$ for every subpartition $\mathcal{P}$ of V.*

If F is a digraph, then Theorem 9 reduces to Theorem 2.

Theorem 9 can be generalized for mixed hypergraphs.

Theorem 10 (Hörsch, Szigeti [13]) *Let $\mathcal{F} = (V, \mathcal{E} \cup \mathcal{A})$ be a mixed hypergraph, $f, g : V \to \mathbb{Z}_+$ functions, and $k \in \mathbb{Z}_+$. There exists an (f, g)-bounded packing of k spanning mixed hyperarborescences in $\mathcal{F}$ if and only if (1) holds and $e_{\mathcal{E} \cup \mathcal{A}}(\mathcal{P}) \geq k|\mathcal{P}| - \min\{k - f(\overline{\cup\mathcal{P}}), g(\cup\mathcal{P})\}$ for every subpartition $\mathcal{P}$ of V.*

If $\mathcal{F}$ is a mixed graph then Theorem 10 reduces to Theorem 9. Theorem 10 is derived from matroid intersection in [13]. One of the main contribution of this paper is to provide a generalization of Theorem 10 in Subsection 4.2. The new result will be obtained from the theory of generalized polymatroids.

Now a generalization of Theorem 4 for mixed graphs follows. For convenience, we present not the original version of the result but one due to Gao, Yang [11] that fits better to our framework.

Theorem 11 (Matsuoka, Tanigawa [18]) *Let $F = (V, E \cup A)$ be a mixed graph and S a multiset of V. There exists a packing of reachability mixed s-arborescences ($s \in S$) in F if and only if for every strongly connected component C of F and every set $\mathcal{P}$ of subsets of P_F^C such that $Z \cap C \neq \emptyset$ and $e_{E \cup A}(Z - C) = 0$ for every $Z \in \mathcal{P}$ and $Z \cap Z' \cap C = \emptyset$ for every $Z, Z' \in \mathcal{P}$, $e_{E \cup A}(\mathcal{P}) \geq |S_{P_F^C}||\mathcal{P}| - \sum_{Z \in \mathcal{P}} |S_Z|$.*

A common generalization of Theorems 7 and 11 was provided by Gao, Yang [11].

Theorem 12 (Gao, Yang [11]) *Let $F = (V, E \cup A)$ be a mixed graph, S a multiset of V and $\mathsf{M} = (S, r_{\mathsf{M}})$ a matroid. There exists a matroid-reachability-based packing of mixed arborescences in F if and only if for every strongly connected component C of F and every set $\mathcal{P}$ of subsets of P_F^C such that $Z \cap C \neq \emptyset$ and $e_{E \cup A}(Z - C) = 0$ for every $Z \in \mathcal{P}$ and $Z \cap Z' \cap C = \emptyset$ for every $Z, Z' \in \mathcal{P}$,*

$$e_{E \cup A}(\mathcal{P}) \quad \geq \quad r_{\mathsf{M}}(S_{P_F^C})|\mathcal{P}| - \sum_{Z \in \mathcal{P}} r_{\mathsf{M}}(S_Z). \tag{9}$$

For $E = \emptyset$, Theorem 12 reduces to Theorem 7. For the free matroid M, Theorem 12 reduces to Theorem 11. Hörsch, Szigeti [14] pointed out that Theorem 12 holds for mixed hypergraphs. That more general result was proved in [14] by induction using a result on matroid-based packing of mixed hyperbranchings in mixed hypergraphs from [6]. Here we propose another approach to prove Theorem 12. It will be derived from Theorem 15, a new orientation result.

We need a matroid construction for hypergraphs and one for mixed hypergraphs. Given a hypergraph $\mathcal{H} = (V, \mathcal{E})$, let $\boldsymbol{\mathcal{I}_{\mathcal{H}}} = \{\mathcal{Z} \subseteq \mathcal{E} : |V(\mathcal{Z}')| > |\mathcal{Z}'| \text{ for all } \emptyset \neq \mathcal{Z}' \subseteq \mathcal{Z}\}$. Lorea [17] showed that $\mathcal{I}_{\mathcal{H}}$ is the set of independent sets of a matroid $\boldsymbol{\mathsf{M}_{\mathcal{H}}}$ on $\mathcal{E}$, called the *hypergraphic matroid* of the hypergraph $\mathcal{H}$. We also need the *k-hypergraphic matroid* $\boldsymbol{\mathsf{M}^k_{\mathcal{H}}}$ of $\mathcal{H}$ which is the k-sum matroid of $\mathsf{M}_{\mathcal{H}}$, that is the matroid on ground set $\mathcal{E}$ in which a subset of $\mathcal{E}$ is independent if it can be partitioned into k independent sets of $\mathsf{M}_{\mathcal{H}}$. Hörsch-Szigeti [13] extend the previous construction to mixed hypergraphs. Let $\mathcal{F} = (V, \mathcal{A} \cup \mathcal{E})$ be a mixed hypergraph. For a subpartition $\mathcal{P}$ of V, $\boldsymbol{\mathcal{A}(\mathcal{P})}$ and $\boldsymbol{\mathcal{E}(\mathcal{P})}$ denote the set of dyperedges and the set of hyperedges that enter some member of $\mathcal{P}$. Let $\boldsymbol{\mathcal{H}_{\mathcal{F}}} = (V, \boldsymbol{\mathcal{E}_{\mathcal{A}}} \cup \mathcal{E})$ the underlying hypergraph of $\mathcal{F}$ and $\boldsymbol{\mathcal{D}_{\mathcal{F}}} = (V, \mathcal{A} \cup \mathcal{A}_{\mathcal{E}})$ the *directed extension* of $\mathcal{F}$ where $\boldsymbol{\mathcal{A}_{\mathcal{E}}} = \bigcup_{e \in \mathcal{E}} \mathcal{A}_e$ and for $e \in \mathcal{E}$, $\boldsymbol{\mathcal{A}_e} = \{(e - x, x) : x \in e\}$. The *extended k-hypergraphic matroid* $\boldsymbol{\mathsf{M}^k_{\mathcal{F}}}$ of $\mathcal{F}$ on $\mathcal{A} \cup \mathcal{A}_{\mathcal{E}}$ is obtained from $\mathsf{M}^k_{\mathcal{H}_{\mathcal{F}}}$ by replacing every $e \in \mathcal{E}$ by $|e|$ parallel copies of itself, associating these elements to the dyperedges in $\mathcal{A}_e$ and associating every hyperedge of $\mathcal{E}_{\mathcal{A}}$ to the corresponding dyperedge in $\mathcal{A}$. It is shown in [13] that the rank function of the extended k-hypergraphic matroid $\mathsf{M}^k_{\mathcal{F}}$ satisfies for all $\mathcal{Z} \subseteq \mathcal{A} \cup \mathcal{A}_{\mathcal{E}}$,

$$r_{\mathsf{M}^k_{\mathcal{F}}}(\mathcal{Z}) = \min\{|\mathcal{Z} \cap \mathcal{A}(\mathcal{P})| + |\{e \in \mathcal{E}(\mathcal{P}) : \overline{\mathcal{Z}} \cap \mathcal{A}_e \neq \emptyset\}| + k(|V| - |\mathcal{P}|) : \mathcal{P} \text{ partition of } V\}. \tag{10}$$

Generalized polymatroids were introduced by Frank [8]. For a pair (p, b) of set functions on S for which $p(\emptyset) = b(\emptyset) = 0$, p is supermodular, b is submodular, and $b(X) - p(Y) \geq b(X - Y) - p(Y - X)$ for all $X, Y \subseteq S$, the polyhedron $\boldsymbol{Q(p, b)} = \{x \in \mathbb{R}^S : p(Z) \leq x(Z) \leq b(Z)\ \forall Z \subseteq S\}$ is called a *generalized-polymatroid*, shortly *g-polymatroid*. The Minkowski sum of the n g-polymatroids $Q(p_i, b_i)$ is denoted by $\sum_1^n \boldsymbol{Q(p_i, b_i)}$. For $\alpha, \beta \in \mathbb{R}$, the polyhedron $\boldsymbol{K(\alpha, \beta)} = \{x \in \mathbb{R}^S : \alpha \leq x(S) \leq \beta\}$ is called a *plank*. For more details on generalized polymatroids see [9]. We will need the following results on g-polymatroids.

Theorem 13 (Frank [9]) *The following hold:*

1. *Let $Q(p, b)$ be a g-polymatroid, $K(\alpha, \beta)$ a plank and $M = Q(p, b) \cap K(\alpha, \beta)$.*
   - *(i) $M \neq \emptyset$ if and only if $p \leq b$, $\alpha \leq \beta$, $\beta \geq p(S)$ and $\alpha \leq b(S)$.*
   - *(ii) M is a g-polymatroid.*
   - *(iii) If $M \neq \emptyset$, then $M = Q(p^\alpha_\beta, b^\alpha_\beta)$ with*

$$\boldsymbol{p^\alpha_\beta(Z)} = \max\{p(Z), \alpha - b(S - Z)\}, \quad \boldsymbol{b^\alpha_\beta(Z)} = \min\{b(Z), \beta - p(S - Z)\}. \tag{11}$$

2. *Let $Q(p_1, b_1)$ and $Q(p_2, b_2)$ be two non-empty g-polymatroids and $M = Q(p_1, b_1) \cap Q(p_2, b_2)$.*
   - *(i) $M \neq \emptyset$ if and only if $p_1 \leq b_2$ and $p_2 \leq b_1$.*
   - *(ii) If p_1, b_1, p_2, b_2 are integral and $M \neq \emptyset$, then M contains an integral element.*
3. *Let $Q(p_i, b_i)$ be n non-empty g-polymatroids. Then $\sum_1^n Q(p_i, b_i) = Q(\sum_1^n p_i, \sum_1^n b_i)$.*

4 Main results

4.1 A new orientation result

To prove the new orientation result, Theorem 15, we need a result of Frank, see Theorem 15.4.13 in [9].

Theorem 14 (Frank [9]) *Let $G = (V, E)$ be a graph and h an integer-valued, intersecting supermodular function such that $h(V) = 0$. There exists an orientation $\vec{G} = (V, \vec{E})$ of G such that $d^-_{\vec{E}}(X) \geq h(X)$ for every $X \subseteq V$ if and only if*

$$e_E(\mathcal{P}) \quad \geq \quad \sum_{X \in \mathcal{P}} h(X) \qquad \text{for every subpartition } \mathcal{P} \text{ of } V. \tag{12}$$

We can now extend an orientation theorem which is implicitly contained in Gao, Yang [11] as follows.

Theorem 15 *Let $F = (V, E \cup A)$ be a mixed graph and b a submodular function on V. There exists an orientation $\vec{E}$ of E such that in $\vec{F} = (V, \vec{E} \cup A)$*

$$d^-_{\vec{E} \cup A}(X) \quad \geq \quad b(P^X_F) - b(X) \qquad \text{for every } X \subseteq V \tag{13}$$

if and only if for every strongly connected component C of F and every set $\mathcal{P}$ of subsets of P^C_F such that $Z \cap C \neq \emptyset$ and $e_{E \cup A}(Z - C) = 0$ for every $Z \in \mathcal{P}$ and $Z \cap Z' \cap C = \emptyset$ for every $Z, Z' \in \mathcal{P}$,

$$e_{E \cup A}(\mathcal{P}) \quad \geq \quad b(P^C_F)|\mathcal{P}| - \sum_{Z \in \mathcal{P}} b(Z). \tag{14}$$

Proof: Let $(\boldsymbol{F} = (\boldsymbol{V}, \boldsymbol{E} \cup \boldsymbol{A}), \boldsymbol{b})$ be a minimum counterexample for Theorem 15. Let $\boldsymbol{C}$ be a strongly connected component of F such that $e_{E \cup A}(\overline{C}) = 0$. Let $\boldsymbol{F'} = (\boldsymbol{C}, \boldsymbol{E'} \cup \boldsymbol{A'})$ be the subgraph of F induced by C and $(\boldsymbol{F''} = (\boldsymbol{V''}, \boldsymbol{E''} \cup \boldsymbol{A''}), \boldsymbol{b''})$ be the instance obtained from (F, b) by deleting the elements in C. As $e_{E \cup A}(\overline{C}) = 0$, we have $e_{E'' \cup A''}(X) = e_{E \cup A}(X)$, $P^X_{F''} = P^X_F$ and $b''(X) = b(X)$ for every $X \subseteq V''$. Then, since (F, b) satisfies (14), so does (F'', b''). Hence, by the minimality of (F, b), there exists an orientation $\vec{\boldsymbol{E}}\boldsymbol{''}$ of E'' such that

$$b(X) + d^-_{\vec{E''} \cup A''}(X) \quad \geq \quad b(P^X_F) \qquad \text{for every } X \subseteq V''. \tag{15}$$

Let $\boldsymbol{b'}(X) = \min\{b(Y) + d^-_A(Y) : Y \subseteq P^C_F, Y \cap C = X, e_{E \cup A}(Y - C) = 0\}$ for every $X \subseteq C$. For any set $X_i \subseteq C$, let $\boldsymbol{Y_i}$ be a set that provides $b'(X_i)$. Gao, Yang [11] proved that b' is submodular. Indeed, for $X_1, X_2 \subseteq C$, let $\boldsymbol{X_3} = X_1 \cap X_2$, $\boldsymbol{X_4} = X_1 \cup X_2$, $\boldsymbol{Y_3} = Y_1 \cap Y_2$ and $\boldsymbol{Y_4} = Y_1 \cup Y_2$. Then, for $i = 3, 4$, we have $Y_i \subseteq P^C_F, Y_i \cap C = X_i$ and $e_{E \cup A}(Y_i - C) = 0$. Then, since b and d^-_A are submodular, so is b'.

Let $\boldsymbol{h}$ be defined by $h(X) = b(P^C_F) - b'(X)$ for every $X \subseteq C$. By the previous claim, h is intersecting supermodular. Let $\boldsymbol{\mathcal{P}} = \{X_1, \ldots, X_t\}$ be a subpartition of C. Let $\boldsymbol{\mathcal{P}'} = \{Y_i : X_i \in \mathcal{P}\}$. Then $\mathcal{P}'$ is a set of subsets of P^C_F such that $Y_i \cap C \neq \emptyset$ and $e_{E \cup A}(Y_i - C) = 0$ for $1 \leq i \leq t$ and $Y_i \cap Y_j \cap C = \emptyset$ for $1 \leq i < j \leq t$. It follows, by (14), that $e_{E'}(\mathcal{P}) = e_{E \cup A}(\mathcal{P}') - e_A(\mathcal{P}) \geq b(P^C_F)|\mathcal{P}'| - \sum_1^t b(Y_i) - \sum_1^t d^-_A(X_i) = \sum_1^t (b(P^C_F) - b(Y_i) - d^-_A(Y_i)) = \sum_1^t h(X_i)$. Thus the graph (C, E') satisfies (12). In particular, we get that $0 = e_{E'}(C) \geq h(C)$. Moreover, $h(C) = b(P^C_F) - b'(C) \geq b(P^C_F) - b(P^C_F) = 0$. Hence $h(C) = 0$. Then, by Theorem 14, there exists an orientation $\vec{\boldsymbol{E}}\boldsymbol{'}$ of E' such that $d^-_{\vec{E'}}(X) \geq h(X) = b(P^C_F) - b'(X)$ for every $X \subseteq C$. It follows that for every $Y \subseteq P^C_F$ with $Y \cap C \neq \emptyset$ and $e_{E \cup A}(Y - C) = 0$, we have

$$d^-_{\vec{E'}}(Y) = d^-_{\vec{E'}}(Y \cap C) \geq b(P^C_F) - b(Y) - d^-_A(Y). \tag{16}$$

Let $\vec{\boldsymbol{F}} = (V, \vec{E} \cup A)$, where $\vec{E} = \vec{E'} \cup \vec{E''}$. To finish the proof we show that $\vec{F}$ satisfies (13). If $X \subseteq V''$, then, by (15), (13) holds. Otherwise, $X \cap C \neq \emptyset$. Let $\boldsymbol{Z} = P^{X-C}_F$, $\boldsymbol{Y} = Z \cap P^C_F$ and $\boldsymbol{W} = Y \cup (X \cap C)$. Then $X \cap Z = X - C, P^C_F \cap (X \cup Z) = W$ and $P^C_F \cup (X \cup Z) = P^X_F$, $e_{E \cup A}(Y) = 0$. Thus, by (15) for $X - C$, (16) for W and the submodularity of b, we have $d^-_{\vec{E} \cup A}(X) \geq d^-_{\vec{E''} \cup A}(X - C) + d^-_{\vec{E'} \cup A}(W) \geq (b(Z) - b(X - C)) + (b(P^C_F) - b(W)) \geq (b(X \cup Z) - b(X)) + (b(P^X_F) - b(X \cup Z)) \geq b(P^X_F) - b(X)$, so (13) holds. □

Theorem 12 easily follows from Theorems 6 and 15. Let (F, S, M) be an instance of Theorem 12 that satisfies (9). Then, by Theorems 15 applied for $b(X) = r_{\mathsf{M}}(S_X)$, there exists an orientation $\vec{E}$ of E

such that in $\vec{F} = (V, \vec{E} \cup A)$ (13) holds. Let $X \subseteq V$. Since $P^X_{\vec{F}} \subseteq P^X_F$ and r_{M} is non-decreasing, we have $r_{\mathsf{M}}(S_{P^X_{\vec{F}}}) \leq r_{\mathsf{M}}(S_{P^X_F})$. By (13) applied for $P^X_{\vec{F}}$, we have $r_{\mathsf{M}}(S_{P^X_{\vec{F}}}) \geq r_{\mathsf{M}}(S_{P^X_F})$. Hence $r_{\mathsf{M}}(S_{P^X_{\vec{F}}}) = r_{\mathsf{M}}(S_{P^X_F})$. Thus (13) implies that (6) holds in $(\vec{F}, S, \mathsf{M})$. Then, by Theorems 6, there exists a matroid-reachability-based packing of arborescences in $(\vec{F}, S, \mathsf{M})$. Since $r_{\mathsf{M}}(S_{P^X_{\vec{F}}}) = r_{\mathsf{M}}(S_{P^X_F})$, by replacing the arcs in $\vec{E}$ by the edges in E, we obtain a matroid-reachability-based packing of mixed arborescences in (F, S, M).

4.2 A new result on packing mixed hyperarborescences

The main contribution of the present paper is a common generalization of Theorems 3 and 10.

Theorem 16 *Let $\mathcal{F} = (V, \mathcal{E} \cup \mathcal{A})$ be a mixed hypergraph, $f, g : V \to \mathbb{Z}_+$ functions, and $k, \ell, \ell' \in \mathbb{Z}_+ - \{0\}$. There exists an (f, g)-bounded k-regular (ℓ, ℓ')-limited packing of mixed hyperarborescences in $\mathcal{F}$ if and only if* (3) *and* (4) *hold and*

$$e_{\mathcal{E}\cup\mathcal{A}}(\mathcal{P}) \quad \geq \quad k|\mathcal{P}| - \min\{\ell' - f(\overline{\cup\mathcal{P}}), g_k(\cup\mathcal{P})\} \qquad \text{for every subpartition } \mathcal{P} \text{ of } V. \tag{17}$$

If $\mathcal{F}$ is a digraph, then Theorem 16 reduces to Theorem 3. If $k = \ell = \ell'$, then Theorem 16 reduces to Theorem 10. Theorem 16 will follow from Theorem 17.

Theorem 17 *Let $\mathcal{F} = (V, \mathcal{E} \cup \mathcal{A})$ be a mixed hypergraph, $f, g : V \to \mathbb{Z}_+$ functions, and $k, \ell, \ell' \in \mathbb{Z}_+ - \{0\}$. Let $\mathsf{M}_{\boldsymbol{v}} = (\rho_{\mathcal{A}\cup\mathcal{A}_\mathcal{E}}(v), r_v)$ be the free matroid for all $v \in V$ and $\mathsf{M}^{\boldsymbol{k}}_{\boldsymbol{\mathcal{F}}}$ the extended k-hypergraphic matroid of $\mathcal{F}$ on $\mathcal{A} \cup \mathcal{A}_\mathcal{E}$. Let $\boldsymbol{T} = (\sum_{v\in V}(Q(0, r_v) \cap K(k - g_k(v), k - f(v)))) \cap K(k|V| - \ell', k|V| - \ell) \cap Q(0, r_{\mathsf{M}^k_\mathcal{F}})$.*

(a) The characteristic vectors of the dyperedge sets of the (f, g)-bounded k-regular (ℓ, ℓ')-limited packings of hyperarborescences in orientations of $\mathcal{F}$ are exactly the integer points of T.

(b) $T \neq \emptyset$ if and only if (3) *and* (4) *hold and for every $\mathcal{Z} \subseteq \mathcal{A} \cup \mathcal{A}_\mathcal{E}$,*

$$\sum_{v\in V} \max\{0, k - g_k(v) - d^-_{\mathcal{Z}}(v)\} \quad \leq \quad r_{\mathsf{M}^k_\mathcal{F}}(\overline{\mathcal{Z}}), \tag{18}$$

$$k|V| - \ell' - \sum_{v\in V} \min\{d^-_{\mathcal{Z}}(v), k - f(v)\} \quad \leq \quad r_{\mathsf{M}^k_\mathcal{F}}(\overline{\mathcal{Z}}). \tag{19}$$

(c) (18) *and* (19) *are equivalent to* (17).

Proof: (a) To prove the **necessity**, let $\boldsymbol{\mathcal{B}_1, \ldots, \mathcal{B}_{\ell^*}}$ be an (f, g)-bounded k-regular packing of hyperarborescences in an orientation $\vec{\boldsymbol{\mathcal{F}}}$ of $\mathcal{F}$, where $\ell \leq \ell^* \leq \ell'$. Let $\boldsymbol{S}$ be the root set of the hyperarborescences in the packing. Note that $|S| = \ell^*$. Let $\boldsymbol{\mathcal{Z}}$ be the dyperedge set of the packing. Since the packing is k-regular, we have $k = d^-_{\mathcal{Z}}(v) + |S_v|$ for all $v \in V$. Then $k|V| = |\mathcal{Z}| + |S|$. Since the packing is (f, g)-bounded, we have $f(v) \leq |S_v| \leq g_k(v)$ for all $v \in V$. Let $\boldsymbol{m}$ be the characteristic vector of $\mathcal{Z}$ and $\boldsymbol{m_v}$ the restriction of m on $\rho_{\mathcal{A}\cup\mathcal{A}_\mathcal{E}}(v)$ for all $v \in V$. Then m_v is a characteristic vector, so $m_v \in Q(0, r_v)$ for all $v \in V$. Since for all $v \in V$, $d^-_{\mathcal{Z}}(v) = m_v(\rho_{\mathcal{A}\cup\mathcal{A}_\mathcal{E}}(v))$, we have $m_v \in K(k - g_k(v), k - f(v))$. It follows that $m \in \sum_{v\in V}(Q(0, r_v) \cap K(k - g_k(v), k - f(v)))$. Since $\ell \leq |S| \leq \ell'$, $k|V| = |\mathcal{Z}| + |S|$ and $|\mathcal{Z}| = m(\mathcal{A} \cup \mathcal{A}_\mathcal{E})$, we have $m \in K(k|V| - \ell', k|V| - \ell)$.

To prove the **sufficiency**, let $\boldsymbol{m} = (m_v)_{v\in V}$ be an integer point of T, that is $m_v \in Q(0, r_v) \cap K(k - g_k(v), k - f(v))$ for all $v \in V$ and $m \in K(k|V| - \ell', k|V| - \ell) \cap Q(0, r_{\mathsf{M}^k_\mathcal{F}})$. Since m_v is an integer point in $Q(0, r_v)$, m_v is the characteristic vector of a subset $\boldsymbol{\vec{\mathcal{Z}}_v}$ of $\rho_{\mathcal{A}\cup\mathcal{A}_\mathcal{E}}(v)$. Since $m_v \in K(k - g_k(v), k - f(v))$, we have $k - g_k(v) \leq m_v(\rho_{\mathcal{A}\cup\mathcal{A}_\mathcal{E}}(v)) = |\vec{\mathcal{Z}}_v| = m_v(\rho_{\mathcal{A}\cup\mathcal{A}_\mathcal{E}}(v)) \leq k - f(v)$. Let $\boldsymbol{\vec{\mathcal{Z}}} = \bigcup_{v\in V} \vec{\mathcal{Z}}_v$. Note that $d^-_{\vec{\mathcal{Z}}}(v) = |\vec{\mathcal{Z}}_v|$ for all $v \in V$. Then, by $f \geq 0$, we have $k - d^-_{\vec{\mathcal{Z}}}(v) \geq f(v) \geq 0$ for all $v \in V$. Since $m \in K(k|V| - \ell', k|V| - \ell)$, we have $k|V| - \ell' \leq m(\mathcal{A} \cup \mathcal{A}_\mathcal{E}) = |\vec{\mathcal{Z}}| = m(\mathcal{A} \cup \mathcal{A}_\mathcal{E}) \leq k|V| - \ell$. Since $m \in Q(0, r_{\mathsf{M}^k_\mathcal{F}})$, we get that $\vec{\mathcal{Z}}$ is independent in $\mathsf{M}^k_\mathcal{F}$. It follows that $\vec{\mathcal{Z}}$ is a subset of the dyperedge set

of an orientation $\vec{\boldsymbol{\mathcal{F}}}$ of $\mathcal{F}$ and for all $X \subseteq V$, $|\mathcal{Z}(X)| \leq r_{\mathsf{M}^k_{\mathcal{H}}}(\mathcal{Z}(X)) \leq k(|X|-1)$ for the hypergraph $\boldsymbol{\mathcal{H}} = (V, \mathcal{E}_{\mathcal{A}} \cup \mathcal{E})$. Let $\boldsymbol{S}$ be the multiset of V where every vertex v is chosen $k - d^-_{\vec{\mathcal{Z}}}(v)$ times, that is $|S_v| = k - d^-_{\vec{\mathcal{Z}}}(v)$ for all $v \in V$. Since $d^-_{\vec{\mathcal{Z}}} \geq 0$, (8) holds for the matroid M on S where the independent sets are the sets of size at most k of S. Since $d^-_{\vec{\mathcal{Z}}}(X) = \sum_{v\in X} d^-_{\vec{\mathcal{Z}}}(v) - |\vec{\mathcal{Z}}(X)| = \sum_{v\in X}(k - |S_v|) - |\mathcal{Z}(X)| \geq k|X| - |S_X| - k(|X|-1) = k - |S_X|$, (7) holds for $\vec{\boldsymbol{\mathcal{F}}}' = (V, \vec{\mathcal{Z}})$ and M. Then, by Theorem 8, there exists a k-regular packing of s-hyperarborescences ($s \in S$) in $\vec{\mathcal{F}}'$ and hence in $\vec{\mathcal{F}}$. Since the number of dyperedges in the packing is $k|V| - |S| = \sum_{v\in V}(k - |S_v|) = \sum_{v\in V} d^-_{\vec{\mathcal{Z}}}(v) = |\vec{\mathcal{Z}}|$, the dyperedge set of the packing is $\vec{\mathcal{Z}}$. Since $f(v) \leq k - |\vec{\mathcal{Z}}_v| = k - d^-_{\vec{\mathcal{Z}}}(v) = |S_v| = k - d^-_{\vec{\mathcal{Z}}}(v) = k - |\vec{\mathcal{Z}}_v| \leq g_k(v) \leq g(v)$ for all $v \in V$, the packing is (f,g)-bounded. Since $\ell \leq k|V| - |\vec{\mathcal{Z}}| = |S| = k|V| - |\vec{\mathcal{Z}}| \leq \ell'$, the number of hyperarborescences in the packing is at least ℓ and at most ℓ'. Finally, since $\vec{\mathcal{F}}$ is an orientation of $\mathcal{F}$, the proof is complete.

(b) By Theorem 13.1, for all $v \in V$, $Q(0, r_v) \cap K(k - g_k(v), k - f(v)) \neq \emptyset$ if and only if $k - g_k(v) \leq k - f(v)$ that is (3) holds and $0 \leq k - f(v)$ (that holds by the previous inequality) and $k - g_k(v) \leq d^-_{\mathcal{A}\cup\mathcal{A}_{\mathcal{E}}}(v)$. Then $Q(0, r_v) \cap K(k - g_k(v), k - f(v)) = Q(p_v, b_v)$ where, by (11), we have for all $\mathcal{Z} \subseteq \mathcal{A} \cup \mathcal{A}_{\mathcal{E}}$,

$$\boldsymbol{p_v}(\mathcal{Z}_v) = \max\{0, k - g_k(v) - d^-_{\overline{\mathcal{Z}}_v}(v)\}, \quad \boldsymbol{b_v}(\mathcal{Z}_v) = \min\{d^-_{\mathcal{Z}_v}(v), k - f(v)\}. \tag{20}$$

By Theorem 13.3, $\sum_{v\in V} Q(p_v, b_v) = Q(p_\Sigma, b_\Sigma)$ where $\boldsymbol{p_\Sigma} = \sum_{v\in V} p_v$, $\boldsymbol{b_\Sigma} = \sum_{v\in V} b_v$. By Theorem 13.1, $Q(p_\Sigma, b_\Sigma) \cap K(k|V| - \ell', k|V| - \ell) \neq \emptyset$ if and only if $Q(p_v, b_v) \neq \emptyset$ for all $v \in V$, $k|V| - \ell' \leq k|V| - \ell$ (which is equivalent to one of the conditions in (4)), $p_\Sigma(\mathcal{A} \cup \mathcal{A}_{\mathcal{E}}) \leq k|V| - \ell$ (which is equivalent to the other condition in (4)) and $b_\Sigma(\mathcal{A} \cup \mathcal{A}_{\mathcal{E}}) \geq k|V| - \ell'$. Then the intersection is equal to $Q(p, b)$ where, by (11), (20), $p_\Sigma = \sum_{v\in V} p_v$, and $b_\Sigma = \sum_{v\in V} b_v$, we have for all $\mathcal{Z} \subseteq \mathcal{A} \cup \mathcal{A}_{\mathcal{E}}$,

$$\boldsymbol{p}(\mathcal{Z}) = \max\left\{\sum_{v\in V} \max\{0, k - g_k(v) - d^-_{\overline{\mathcal{Z}}_v}(v)\}, k|V| - \ell' - \sum_{v\in V} \min\{d^-_{\overline{\mathcal{Z}}_v}(v), k - f(v)\}\right\}, \tag{21}$$

$$\boldsymbol{b}(\mathcal{Z}) = \min\left\{\sum_{v\in V} \min\{d^-_{\mathcal{Z}_v}(v), k - f(v)\}, k|V| - \ell - \sum_{v\in V} \max\{0, k - g_k(v) - d^-_{\mathcal{Z}_v}(v)\}\right\}. \tag{22}$$

By Theorem 13.2, $Q(p, b) \cap Q(0, r_{\mathsf{M}^k_{\mathcal{F}}}) \neq \emptyset$ if and only if $Q(p, b) \neq \emptyset$, $p \leq r_{\mathsf{M}^k_{\mathcal{F}}}$ which, by (21), is equivalent to (18) and (19), and $b \geq 0$ (which holds by $b \geq p \geq 0$). Note that $k - g_k(v) \leq d^-_{\mathcal{A}\cup\mathcal{A}_{\mathcal{E}}}(v)$ for all $v \in V$ and $b_\Sigma(\mathcal{A} \cup \mathcal{A}_{\mathcal{E}}) \geq k|V| - \ell'$ follow from $p \leq r_{\mathsf{M}^k_{\mathcal{F}}}$ applied for $\mathcal{Z} = \emptyset$ and the proof is complete.

(c) We note that (18) is equivalent to

$$k|V| - g_k(V) - \sum_{v\in V} \min\{d^-_{\mathcal{Z}}(v), k - g_k(v)\} \leq r_{\mathsf{M}^k_{\mathcal{F}}}(\overline{\mathcal{Z}}). \tag{23}$$

First we show that (18) and (19) imply (17). Let $\boldsymbol{\mathcal{P}}$ be a subpartition of V. Let $\boldsymbol{\mathcal{Z}} = \bigcup_{v\in\overline{\cup\mathcal{P}}} \rho_{\mathcal{A}}(v) \cup \bigcup_{e\in\mathcal{E}(\mathcal{F}(\overline{\cup\mathcal{P}}))} \mathcal{A}_e$ and $\boldsymbol{\mathcal{P}'} = \mathcal{P} \cup \{v\}_{v\in\overline{\cup\mathcal{P}}}$. Note that $d^-_{\mathcal{Z}}(v) = 0$ for all $v \in \cup\mathcal{P}$,

$$\sum_{v\in V} \min\{d^-_{\mathcal{Z}}(v), k - h(v)\} \leq k|\overline{\cup\mathcal{P}}| - h(\overline{\cup\mathcal{P}}) \text{ for } h \in \{g_k, f\}, \tag{24}$$

$\mathcal{P}'$ is a partition of V, and, by (10),

$$r_{\mathsf{M}^k_{\mathcal{F}}}(\overline{\mathcal{Z}}) \leq |\overline{\mathcal{Z}} \cap \mathcal{A}(\mathcal{P}')| + |\{e \in \mathcal{E}(\mathcal{P}') : \overline{\mathcal{Z}} \cap \mathcal{A}_e \neq \emptyset\}| + k(|V| - |\mathcal{P}'|) = e_{\mathcal{A}\cup\mathcal{A}_{\mathcal{E}}}(\mathcal{P}) + k(|V| - |\mathcal{P}| - |\overline{\cup\mathcal{P}}|). \tag{25}$$

Then (23), (24) applied for $h = g_k$ and (25) imply $e_{\mathcal{E}\cup\mathcal{A}}(\mathcal{P}) \geq k|\mathcal{P}| - g_k(\cup\mathcal{P})$. Similarly, (19), (24) applied for $h = f$ and (25) imply $e_{\mathcal{E}\cup\mathcal{A}}(\mathcal{P}) \geq k|\mathcal{P}| - \ell' + f(\overline{\cup\mathcal{P}})$. Hence (17) follows.

We now show that (17) implies (19) and (23) and hence (18). Let $\boldsymbol{\mathcal{Z}} \subseteq \mathcal{A} \cup \mathcal{A}_{\mathcal{E}}$. By (10), there exists a partition $\boldsymbol{\mathcal{P}}$ of V such that for $\boldsymbol{\mathcal{K}} = \{e \in \mathcal{E}(\mathcal{P}) : \overline{\mathcal{Z}} \cap \mathcal{A}_e \neq \emptyset\}$, we have

$$r_{\mathsf{M}^k_{\mathcal{F}}}(\overline{\mathcal{Z}}) = |\overline{\mathcal{Z}} \cap \mathcal{A}(\mathcal{P})| + |\mathcal{K}| + k(|V| - |\mathcal{P}|). \tag{26}$$

For $h \in \{g_k, f\}$, let $\boldsymbol{\mathcal{P}_h} = \{X \in \mathcal{P} : d^-_{\mathcal{Z}}(v) \leq k - h(v) \text{ for all } v \in X\}$. Note that $\mathcal{P}_h$ is a subpartition of V and for every $X \in \mathcal{P} - \mathcal{P}_h$, there exists a vertex $v_X \in X$ such that $d^-_{\mathcal{Z}}(v_X) > k - h(v_X)$. By the definition of $\mathcal{K}$, we have

$$\mathcal{A}_{\mathcal{E}(\mathcal{P}_h)-\mathcal{K}} \subseteq \mathcal{Z} \cap \mathcal{A}_{\mathcal{E}(\mathcal{P}_h)}. \tag{27}$$

Then, by (26), the definition of $\mathcal{P}_h$, the definition of v_X, $d^-_{\mathcal{Z}} \geq 0, k - h \geq 0$, (27), and $h \geq 0$, we have

$$\begin{aligned}
& r_{\mathsf{M}^k_{\mathcal{F}}}(\overline{\mathcal{Z}}) + \sum_{v \in V} \min\{d^-_{\mathcal{Z}}(v), k - h(v)\} \\
= \; & |\overline{\mathcal{Z}} \cap \mathcal{A}(\mathcal{P})| + |\mathcal{K}| + k(|V| - |\mathcal{P}|) + \sum_{v \in \cup\mathcal{P}_h} \min\{d^-_{\mathcal{Z}}(v), k - h(v)\} + \sum_{v \in \overline{\cup\mathcal{P}_h}} \min\{d^-_{\mathcal{Z}}(v), k - h(v)\} \\
\geq \; & |\overline{\mathcal{Z}} \cap \mathcal{A}(\mathcal{P}_h)| + \sum_{v \in \cup\mathcal{P}_h} d^-_{\mathcal{Z}}(v) + \sum_{X \in \mathcal{P} - \mathcal{P}_h} \sum_{v \in X} \min\{d^-_{\mathcal{Z}}(v), k - h(v)\} + |\mathcal{K}| + k(|V| - |\mathcal{P}|) \\
\geq \; & |\overline{\mathcal{Z}} \cap \mathcal{A}(\mathcal{P}_h)| + |\mathcal{Z} \cap \mathcal{A}(\mathcal{P}_h)| + |\mathcal{Z} \cap \mathcal{A}_{\mathcal{E}(\mathcal{P}_h)}| + \sum_{X \in \mathcal{P} - \mathcal{P}_h} (k - h(v_X)) + |\mathcal{K}| + k(|V| - |\mathcal{P}|) \\
\geq \; & |\mathcal{A}(\mathcal{P}_h)| + |\mathcal{A}_{\mathcal{E}(\mathcal{P}_h)-\mathcal{K}}| + \sum_{X \in \mathcal{P} - \mathcal{P}_h} (k - h(X)) + |\mathcal{K}| + k(|V| - |\mathcal{P}|) \\
\geq \; & e_{\mathcal{E} \cup \mathcal{A}}(\mathcal{P}_h) - |\mathcal{K}| + k(|\mathcal{P}| - |\mathcal{P}_h|) - h(\overline{\cup\mathcal{P}_h}) + |\mathcal{K}| + k(|V| - |\mathcal{P}|) \\
\geq \; & e_{\mathcal{E} \cup \mathcal{A}}(\mathcal{P}_h) - k|\mathcal{P}_h| - h(\overline{\cup\mathcal{P}_h}) + k|V|.
\end{aligned}$$

The above inequality applied for $h = f$ and (17) provide that $r_{\mathsf{M}^k_{\mathcal{F}}}(\overline{\mathcal{Z}}) + \sum_{v \in V} \min\{d^-_{\mathcal{Z}}(v), k - h(v)\} \geq k|V| - \ell'$, so (19) holds. Similarly, the above inequality applied for $h = g_k$ and (17) provide that $r_{\mathsf{M}^k_{\mathcal{F}}}(\overline{\mathcal{Z}}) + \sum_{v \in V} \min\{d^-_{\mathcal{Z}}(v), k - h(v)\} \geq k|V| - g_k(V)$, so (23) holds. The proof of the theorem is complete. □

We finish the paper by showing that Theorems 13 and 17 imply Theorem 16. Let $(\mathcal{F} = (V, \mathcal{E} \cup \mathcal{A}), f, g, k, \ell, \ell')$ be an instance of Theorem 16 that satisfies (3), (4) and (17). Since (17) holds, by Theorem 17(c), (18) and (19) hold. Since (3) and (4) also hold, by Theorem 17(b), the polyhedron T, defined in Theorem 17, is not empty. Then, by Theorem 13.2(ii), T contains an integral element x. By Theorem 17(b), x is the characteristic vector of the dyperedge set of an (f, g)-bounded k-regular (ℓ, ℓ')-limited packing of hyperarborescences in an orientation $\vec{F} = (V, \vec{\mathcal{E}} \cup \mathcal{A})$ of $\mathcal{F}$. By replacing the arcs in $\vec{\mathcal{E}}$ by the hyperedges in $\mathcal{E}$, we obtain the required packing.

5 Acknowledgements

I thank an anonymous referee for his valuable suggestions that improved the presentation of the paper. I also thank Pierre Hoppenot for his very careful reading of the paper.

References

[1] K. Bérczi, A. Frank, Variations for Lovász' submodular ideas, *in Building Bridges, Springer,* (2008) 137–164.

[2] K. Bérczi, A. Frank, Supermodularity in Unweighted Graph Optimization I: Branchings and Matchings, *Math. Oper. Res.* **43**(3) (2018) 726–753.

[3] M. C. Cai, Arc-disjoint arborescences of digraphs, *J. Graph Theory* **7** (1983) 235–240.

[4] O. Durand de Gevigney, V. H. Nguyen, Z. Szigeti, Matroid-Based Packing of Arborescences, *SIAM J. Discret. Math.* **27**(1) (2013) 567–574.

[5] J. Edmonds, Edge-disjoint branchings, *Combinatorial Algorithms, B. Rustin ed., Academic Press, New York,* (1973) 91–96.

[6] Q. Fortier, Cs. Király, M. Léonard, Z. Szigeti, A. Talon, Old and new results on packing arborescences, *Discret. Appl. Math.* **242** (2018) 26–33.

[7] A. Frank, On disjoint trees and arborescences, *In Algebraic Methods in Graph Theory, 25, Colloquia Mathematica Soc. J. Bolyai, Norh-Holland,* (1978) 59–169.

[8] A. Frank, Generalized polymatroids, *in: A. Hajnal et. al. eds. Finite and infinite sets,* North-Holland, Amsterdam-New York (1984) 285–294.

[9] A. Frank, Connections in Combinatorial Optimization, *Oxford University Press*, 2011.

[10] A. Frank, T. Király, Z. Király, On the orientation of graphs and hypergraphs, *Discret. Appl. Math.* **131**(2) (2003) 385–400.

[11] H. Gao, D. Yang, Packing of maximal independent mixed arborescences, *Discret. Appl. Math.* **289** (2021) 313–319.

[12] H. Gao, D. Yang, Packing of spanning mixed arborescences, *J. Graph Theory*, **98**(2) (2021) 367–377.

[13] F. Hörsch, Z. Szigeti, Packing of mixed hyperarborescences with flexible roots via matroid intersection, *Electronic Journal of Combinatorics*, **28** (3) (2021) P3.29.

[14] F. Hörsch, Z. Szigeti, Reachability in arborescence packings, *Discret. Appl. Math.* **320** (2022) 170–183.

[15] N. Kamiyama, N. Katoh, A. Takizawa, Arc-disjoint in-trees in directed graphs, *Comb.* **29** (2009) 197–214.

[16] Cs. Király, On maximal independent arborescence packing, *SIAM J. Discret. Math.* **30**(4) (2016) 2107–2114.

[17] M. Lorea, Hypergraphes et matroides, *Cahiers Centre Etudes Rech. Oper.* **17** (1975) 289-291.

[18] T. Matsuoka, S. Tanigawa, On Reachability Mixed Arborescences Packing, *Discret. Optim.* **32** (2019) 1–10.

[19] Z. Szigeti, A survey on packing arborescences, in preparation

Quantum-Relaxation Based Optimization Algorithms: Theoretical Extensions

Kosei Teramoto

Department of Computer Science,
The University of Tokyo
teramoto@is.s.u-tokyo.ac.jp

Rudy Raymond

IBM Quantum, IBM Japan
Dept. of Computer Science, The Univ. of Tokyo
Quantum Computing Center, Keio University
rudyhar@jp.ibm.com

Eyuri Wakakuwa

Department of Computer Science,
The University of Tokyo
eyuriwakakuwa@is.s.u-tokyo.ac.jp

Hiroshi Imai

Department of Computer Science,
The University of Tokyo
imai@is.s.u-tokyo.ac.jp

Abstract: Quantum Random Access Optimizer (QRAO) is a quantum-relaxation based optimization algorithm proposed by Fuller et al. that utilizes Quantum Random Access Code (QRAC) to encode multiple variables of binary optimization in a single qubit. The approximation ratio bound of QRAO for the maximum cut problem is 0.555 if the bit-to-qubit compression ratio is 3x, while it is 0.625 if the compression ratio is 2x, thus demonstrating a trade-off between space efficiency and approximability. In this research, we extend the quantum-relaxation by using another QRAC which encodes three classical bits into two qubits (the bit-to-qubit compression ratio is 1.5x) and obtain its approximation ratio for the maximum cut problem as 0.722. Also, we design a novel quantum-relaxation that always guarantees a 2x bit-to-qubit compression ratio which is unlike the original quantum relaxation of Fuller et al. We analyze the condition when it has a non-trivial approximation ratio bound $\left(> \frac{1}{2}\right)$. We hope that our results lead to the analysis of the quantum approximability and practical efficiency of the quantum-relaxation based approaches.

Keywords: Quantum-Relaxation, Quantum Random Access Codes, Quantum State Rounding, Maximum Cut Problem, Quantum Approximability

1 Introduction

1.1 Backgrounds

Solving optimization problems is one of the most important tasks for which quantum computation is expected to be useful. Various quantum algorithms have been devised for NP-hard optimization problems such as QAOA (Quantum Approximate Optimization Algorithms) [3] proposed by Farhi, Goldstone, and Gutmann, and VQE (Variational Quantum Eigensolver) [13] proposed by Peruzzo et al. Although QAOA and VQE are classical-quantum hybrid algorithms designed for near-term devices capable of running only shallow circuits, there are some critical issues. The first issue is scalability. Because QAOA and VQE encode one classical bit into one qubit and the number of qubits of near-term quantum devices is at most several hundred qubits, the problem instance sizes are highly limited. The second issue is that we do not know if *quantumness* (i.e. quantum entanglement) of constant-depth QAOA and VQE can give rise to a better result than the classical optimization algorithms, as indicated in [12]. In other words, for combinatorial optimization, QAOA and VQE may not be attractive to be run on a quantum computer in the first place.

Recently, a new classical-quantum hybrid optimization algorithm, QRAO (Quantum Random Access Optimization) [4] was proposed by Fuller et al. to address the above issues. Specifically, the QRAO encodes multiple classical bits (less than or equal to three) into one qubit using the $(3,1)$-QRAC (Quantum Random Access Code) [1, 6]. Here, (m,n)-QRAC means the quantum random access codes which encode m classical bits into n qubits. Due to this constant-factor improvement in scalability, Fuller et al. were able to perform experiments with QRAO on superconducting quantum devices to solve the largest instances of a maximum cut problem (up to 40 nodes using only 15 qubits). Also, since QRAO searches for quantum states that correspond to solutions to the relaxation problem rather than classical solutions, the quantum state that is eventually discovered is an entangled state that cannot be directly interpreted as a classical solution. Because of this, the methods like QRAO are called *quantum-relaxation* and have been extended for more general quadratic programs [17]. To obtain the classical solution, *quantum state rounding* of the relaxed solution must be performed. Therefore, compared to standard VQE methods, QRAO may benefit from quantum entanglement if the entangled states result in better relaxed values. In other words, QRAO is inherently different from standard quantum-classical hybrid algorithms like QAOA and may benefit from quantum mechanical properties. In fact, there exists an experimental result that there are some instances for which entanglement helps QRAO find optimal solutions [14].

The quantum state rounding algorithm (*magic state rounding*) used in QRAO is inspired by Goemans and Williamson's approximation algorithm for the maximum cut problem with an approximation ratio of 0.879 [5]. It randomly chooses the pair of two-bit-inverted relationships and decodes the encoded bits into one of the two candidates by performing the corresponding quantum measurement. By quantum information theoretic analysis, it is proved that the approximation ratio of quantum-relaxation using $(3,1)$-QRAC is 0.555 and that of quantum-relaxation using $(2,1)$-QRAC is 0.625 [4]. While the optimality of standard QAOA or VQE is often assumed when the obtained quantum state is the ground state, the approximation ratios of QRAO are obtained regardless of the reachability of the ground state. Namely, the ratios are guaranteed as long as the relaxed value of the obtained quantum state exceeds that of the classical optimal value. This is crucial as finding the exact ground state can be extremely hard [9].

The approximation ratios of $(3,1)$- and $(2,1)$-QRAC imply that the higher the space compression ratio the lower the approximation ratio is. There is a trade-off between space efficiency and approximability. The approximation ratio bound of QRAO is much lower than Goemans and Williamson's 0.879 [5] which is proved to be optimal under the UGC (Unique Game Conjecture) [10]. This is because the success probability of decoding each bit of the QRACs used in QRAO is not high. The success probability of decoding each encoded bit is $\frac{1}{2}+\frac{1}{2\sqrt{2}} \approx 0.85$ for $(2,1)$-QRAC and $\frac{1}{2}+\frac{1}{2\sqrt{3}} \approx 0.79$ for $(3,1)$-QRAC [1, 6].

1.2 Our Results

In this research, we extend the quantum-relaxation in two ways: (i) we introduce the use of $(3,2)$-QRAC to obtain a better approximation ratio with a slightly lower bit-to-qubit compression ratio, and (ii) we design a novel quantum-relaxation that always guarantees 2x bit-to-qubit compression ratio which is unlike the original quantum relaxation of Fuller et al. For (i), we will show the formulation of the $(3,2)$-QRAC which encodes three classical bits into two qubits obtained by numerical calculation [8]. The success probability of decoding each encoded bit is $\frac{1}{2}+\frac{1}{\sqrt{6}} \approx 0.908$, and it is optimal among all $(3,2)$-QRACs based on the bound by Manvčinska and Storgaard [11]. Also, we extended the quantum-relaxation by using this $(3,2)$-QRAC. The instance of the problem is encoded into the problem Hamiltonian, and the maximum eigenstate of the Hamiltonian is explored. By performing the quantum state rounding algorithm, we obtain the classical binary solution to the problem. Furthermore, we proved the approximation ratio bound of the above quantum-relaxation based optimization algorithm for the MaxCut problem as $\frac{13}{18} \approx 0.722$. The only assumption of the proof of the approximation ratio is the same as the one using $(3,1)$- or $(2,1)$-QRACs, that is, the energy of the found candidate quantum state for the maximum eigenstate of the problem Hamiltonian exceeds the optimum value of the original problem instance. Although the space compression ratio of our quantum relaxation is $\frac{3}{2} = 1.5$ and is lower than the one using $(3,1)$- or $(2,1)$-QRACs, the approximation ratio bound is better. Our result is consistent with the trade-off between the space compression ratio and the approximability of the maximum cut problem. Though the

obtained approximation ratio bound 0.722 is lower than that of Goemans and Williamson, the practical feasibility of quantum-relaxation based approaches is enhanced.

To always guarantee the bit-to-qubit compression ratio of QRAO using $(3,1)$-QRAC is essential as in the original QRAO the ratio becomes lower as the density of the graph instance increases. This is because there is a constraint that the endpoints of each edge must be associated with different qubits. For example, if the graph instance is the complete graph, then the number of qubits needed to run QRAO is the same as the number of vertices. In such cases, the quantum-relaxation based optimizer has no space advantage against standard QAOA and VQE algorithms. In this research, for (ii), we propose new types of encoding which encode up to two classical bits into a single-qubit by using the $(3,1)$-QRAC. The third encoded bit's position in $(3,1)$-QRAC corresponds to the parity of the two bits. This modification allows us to remove the constraint that the endpoints of each edge have to be assigned to different qubits. The space compression ratio of the algorithm is always 2x which is independent of the density of the graph instances. Unfortunately, non-trivial approximation ratio bound $\left(>\frac{1}{2}\right)$ does not exist generally. We calculate the approximation ratio of this new algorithm by using two parameters ϵ and λ as $\max\left\{\frac{3-2\lambda+2\epsilon}{3+6\epsilon}, \frac{9-2\sqrt{3}+2\sqrt{3}\lambda+2\epsilon}{9+18\epsilon}\right\}$. The parameter ϵ is defined by the equation $\mathrm{OPT} = \left(\frac{1}{2}+\epsilon\right)|E|$ where OPT is the optimal cut value, and therefore ϵ quantifies the so-called MaxCutGain [2]. The parameter λ is the ratio of the edges whose endpoints are assigned to different qubits. By using the approximation ratio bound, we analyze the condition of the graph instance that our algorithm gives a non-obvious approximation ratio bound for the maximum cut problem.

In this paper, we briefly summarize our results. To address more detailed contents, please refer to the paper [15].

2 Preliminaries

2.1 Quantum Random Access Codes

The n qubits are represented by a vector in $\mathbb{C}^{2^n}$ and seem to have much more information than the classical n bits. However, it is known that n qubits are needed to transfer n-bit classical information without error by Holevo bound [7]. On the other hand, if we admit some errors, we can encode multiple classical bits into a single qubit by using $(n,1,p)$-QRA codes [1].

Definition 1 ($(n,1,p)$-QRA codes [1]) *An $(n,1,p)$-QRA coding is a function that maps n-bit strings $x \in \{0,1\}^n$ to 1-qubit states ρ_x satisfying the following conditions that for every $i \in \{1,2,...,n\}$, there exists a POVM*

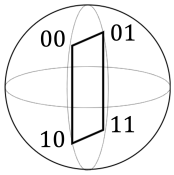
$$E^i = \{E_0^i, E_1^i\}$$

such that

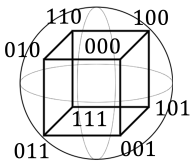
$$\mathrm{Tr}(E_{x_i}^i \rho_x) \geq p$$

for all $x \in \{0,1\}^n$, where x_i is the i-the bit of x.

The POVM E^i corresponds to the decoding process. By measuring the encoded state ρ_x with the POVM E^i, we can decode the i-th encoded bits x_i with probability p. We noted that $(n,1,p)$-QRA codes is meaningless if $p \leq \frac{1}{2}$ because $p = \frac{1}{2}$ is equivalent to randomly choosing binary bits. (n,m,p)-QRA coding for $m \geq 2$ can also be defined in the same way. There exists $(2,1,0.85)$- and $(3,1,0.79)$-QRA codings [1] which are used in QRAO [4]. The $(2,1,0.85)$-QRA coding is visualized as vertices of the square on the x-z plane in the Bloch sphere as shown in Figure 1a. The $(3,1,0.79)$-QRA coding is visualized as vertices of the cube inscribed in the Bloch sphere as shown in Figure 1b.



(a) $(2, 1, 0.85)$-QRA coding



(b) $(3, 1, 0.79)$-QRA coding

Figure 1: The $(n, 1, p)$-QRA coding in Bloch sphere representation

2.2 Quantum Relaxation Based Optimization Algorithms

The following explanation is based on the QRAO paper [4]. We explain the quantum-relaxation based optimization algorithm by using the MaxCut problem formulated as

$$\max_{\{+1,-1\}^{|V(G)|}} \frac{1}{2} \sum_{e_{i,j} \in E(G)} (1 - x_i x_j) \tag{1}$$

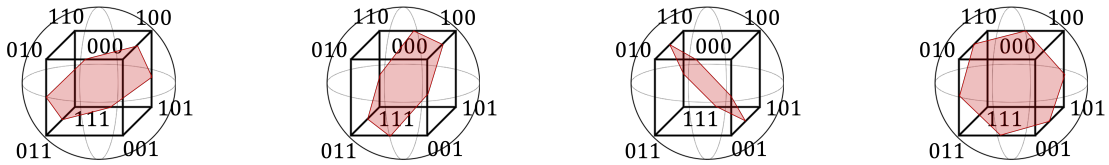
In the typical quantum-classical hybrid approach using variational methods such as VQE [13] or QAOA [3], each classical binary variable x_i is mapped to i-th qubit using the Pauli Z operator. Then the MaxCut problem is reduced to the problem to find the maximum eigenstate of the Hamiltonian:

$$H = \frac{1}{2} \sum_{e_{i,j} \in E(G)} (I - Z_i Z_j). \tag{2}$$

On the other hand, in the quantum-relaxation based optimization algorithms such as QRAO [4], multiple classical bits are encoded into a smaller number of qubits using QRACs explained in Section 2.1. For example, if we use $(3, 1)$-QRAC, three classical binary variables x_1, x_2, and x_3 are mapped to a single qubit using the Pauli X, Y, and Z operators respectively. Compared with QAOA or VQE, QRAO has the constant-factor space complexity advantage. The goal is, as well as the typical methods, to reduce the MaxCut problem to the procedure to explore the maximum eigenstate of the Hamiltonian called *relaxed Hamiltonian* H_{relax}. To construct a relaxed Hamiltonian, we make the mapping from classical binary variables into qubits. First we perform a coloring of the instance graph G by using, for example, LDF (large-degree-first) method [16] whose time complexity is $O(|V(G)| \log |V(G)| + \deg(G)|V(G)|)$ where $\deg(G)$ is the maximum degree of the graph G. After performing the LDF algorithm, the vertices are partitioned into the set $\{V_c\}$ associated with the color $c \in C$. We note that there is a constraint: for each edge, its endpoints must be assigned to different qubits. Next, we associate $\left\lceil \frac{|V_c|}{3} \right\rceil$ qubits for each color $c \in C$. Now up to three vertices are assigned to a single qubit. We greedily order these three vertices and assign the Pauli operators X, Y, and Z respectively. If we use the $(2, 1)$-QRAC, then we associate $\left\lceil \frac{|V_c|}{2} \right\rceil$ qubits for each color and assign the Pauli X and Z for the up to two vertices assigned to the same single qubit instead. Finally, we obtained a relaxed Hamiltonian instead of the *normal* Hamiltonian in Equation (2) as below:

$$H_{relax} = \frac{1}{2} \sum_{e_{i,j} \in E(G)} (I - 3P_i P_j), \tag{3}$$

where P_i is the Pauli operator associated with the vertex v_i. We explore the maximum eigenstate of H_{relax} by using variational methods such as VQE. The relaxed Hamiltonian H_{relax} is no longer diagonal and it contains the non-classical states (with superposition and entanglement) as the maximal eigenstates. It means that the found eigenstate for the relaxed Hamiltonian cannot be associated with the classical solution directly. Because of the construction of the Hamiltonian, the found state should be a quantum state that corresponds to the relaxed solution to the MaxCut problem. A relaxed solution means the solution of the MaxCut problem without the constraint that the solution must be a binary vector. We



(a) $\mu_1^{\pm}$ **(b)** $\mu_2^{\pm}$ **(c)** $\mu_3^{\pm}$ **(d)** $\mu_4^{\pm}$

Figure 2: The intuition of the quantum measurements performed in magic state rounding algorithm

denote the found eigenstate in quantum-relaxation based optimization algorithm as ρ_{relax} and called it *relaxed state*. To retrieve the classical solution for the MaxCut problem, we perform quantum state rounding algorithms. There are two types of rounding algorithms proposed by Fuller et al. [4].

The first rounding algorithm is *Pauli rounding* which decodes the encoded three classical bits in each qubit one by one. Because the Pauli rounding algorithm does not consider the correlation between qubits, if the relaxed state is very entangled, there is no guarantee that the Pauli rounding works well. By using the second rounding algorithm, *magic state rounding*, we can avoid the above problem and can obtain the approximation ratio bound for the MaxCut problem. The idea of the magic state rounding algorithm is to decode three classical variables at once from a single qubit. Consider the single qubit magic state:

$$\mu^{\pm} := \frac{1}{2}\left(I \pm \frac{1}{\sqrt{3}}(X+Y+Z)\right), \tag{4}$$

and set

$$\mu_1^{\pm} := \mu^{\pm}, \mu_2^{\pm} := X\mu^{\pm}X, \mu_3^{\pm} := Y\mu^{\pm}Y, \mu_4^{\pm} := Z\mu^{\pm}Z. \tag{5}$$

In the magic state rounding algorithm, one of the measurement basis $\{\mu_i^+, \mu_i^-\}$ is selected from $i \in [4]$ for each qubit. After choosing the bases for all qubits, then a relaxed state ρ_{relax} is measured on those bases. Three classical binary variables are decoded according to the measurement outcome for each qubit. Figure 2 shows the intuition of the magic state rounding algorithm. Each measurement $\mu_i^{\pm}$ decodes one of the pair of three bits located at opposite angles on the cube (e.g. 000 or 111 in the case of $\mu_1^{\pm}$). By using this simultaneous decoding of the encoded three bits, the magic state rounding algorithm extracts the solution of the MaxCut for every iteration. The magic state rounding algorithm repeats this procedure enough times and outputs the best solution. Unlike the case using the Pauli rounding algorithm, there is an approximation ratio bound for the MaxCut problem when using the magic state rounding algorithm.

Theorem 2 ([4]) *Given access to an oracle $\mathcal{O}_{relax}$ producing relaxed state ρ_{relax} s.t. $\mathrm{Tr}[H_{relax}\rho_{relax}] \geq OPT$, the magic state rounding algorithm produces a solution to the MaxCut problem with expected approximation ratio $\mathbb{E}[\gamma] \geq \frac{5}{9} \approx 0.555$.*

The expected approximation ratio for the QRAO using $(2,1)$-QRAC is proved to be $\frac{5}{8} = 0.625$. In the case of using $(1,1)$-QRAC, the approximation ratio is obtained as 1.0. However, it is meaningless because the existence of the oracle $\mathcal{O}_{relax}$ in the assumption of the proof implies that the oracle can prepare the optimal solution. It is obvious that given the optimum solution, the approximation ratio is 1.0. There is a trade-off between the space compression ratio and the approximation ratio.

3 Theoretical Extensions of Quantum Relaxations

3.1 $(3,2)$-QRA Coding

$(3,2)$-QRA coding is one of the quantum random access codes which encodes three classical bits into two qubits. The concrete formulation of the $(3,2)$-QRAC is obtained in the numerical calculation [8] like the following:

Theorem 3 *Consider the map from three bits* $(x_1, x_2, x_3) \in \{0,1\}^3$ *to a two-qubit quantum state* ρ'_{x_1,x_2,x_3} *defined by the following equations:*

- *If* $b_1 \oplus b_2 \oplus b_3 = 0$,

$$\rho'_{x_1,x_2,x_3} := \frac{1}{4} I_1 I_2 + \frac{1}{4}((-1)^{x_1} Z_1 I_2 + (-1)^{x_2} I_1 Z_2 + (-1)^{x_3} Z_1 Z_2). \tag{6}$$

- *Else if* $b_1 \oplus b_2 \oplus b_3 = 1$,

$$\begin{aligned} \rho'_{x_1,x_2,x_3} :=& \frac{1}{4} I_1 I_2 + (-1)^{x_1} \left(\frac{1}{12} Z_1 I_2 + \frac{1}{6} X_1 X_2 + \frac{1}{6} X_1 Z_2 \right) \\ &+ (-1)^{x_2} \left(\frac{1}{6} I_1 X_2 + \frac{1}{12} I_1 Z_2 + \frac{1}{6} Y_1 Y_2 \right) + (-1)^{x_3} \left(\frac{1}{12} Z_1 Z_2 - \frac{1}{6} X_1 I_2 - \frac{1}{6} Z_1 X_2 \right) \end{aligned} \tag{7}$$

For every pair of (x_1, x_2, x_3), ρ'_{x_1,x_2,x_3} *is a pure state. Then, this map is a* $(3, 2, 0.908)$*-QRA coding with the POVMs (projective measurements, in fact):*

$$F^1 = \left\{ \frac{1}{2} I_1 I_2 \pm \frac{1}{\sqrt{6}} \left(\frac{1}{2} X_1 X_2 + \frac{1}{2} X_1 Z_2 + Z_1 I_2 \right) \right\}, \tag{8}$$

$$F^2 = \left\{ \frac{1}{2} I_1 I_2 \pm \frac{1}{\sqrt{6}} \left(\frac{1}{2} Y_1 Y_2 + \frac{1}{2} I_1 X_2 + I_1 Z_2 \right) \right\}, \tag{9}$$

$$F^3 = \left\{ \frac{1}{2} I_1 I_2 + \frac{1}{\sqrt{6}} \left(Z_1 Z_2 - \frac{1}{2} X_1 I_2 - \frac{1}{2} Z_1 X_2 \right) \right\}. \tag{10}$$

$(3,2)$-QRAC has two kinds of encoded state form in Equations (6) and (7), and which to use depends on the parity of the encoded three bits. It holds that for each parity, four encoded states are orthogonal, i.e. for each $x_1, x_2, x_3 \in \{0,1\}^3$ and $x'_1, x'_2, x'_3 \in \{0,1\}^3$ $((x_1, x_2, x_3) \neq (x'_1, x'_2, x'_3))$ satisfying $x_1 \oplus x_2 \oplus x_3 = x'_1 \oplus x'_2 \oplus x'_3$, $\langle \psi'(x_1, x_2, x_3) | \psi'(x'_1, x'_2, x'_3) \rangle = 0$. It implies that if we know the parity of the encoded classical bits in advance, we can decode the encoded three bits by using the 4-outcome quantum measurement. This characteristic is used when we formulate the rounding algorithm corresponding to the magic state rounding algorithm of the quantum relaxation using $(3,1)$- or $(2,1)$-QRACs. The POVMs in Equations (8) to (10) are used when we'd like to decode the encoded bits one by one (e.g. the Pauli rounding algorithm). The success probability of the decoding is $\frac{1}{2} + \frac{1}{\sqrt{6}} \approx 0.908$, and it is proved to be optimal by using the bound by Mančinska and Storgaard [11]. While the space compression ratio of $(3,2)$-QRAC is less than $(3,1)$- or $(2,1)$-QRACs, the success probability of decryption is better than theirs.

3.2 Quantum Relaxation Using $(3,2)$-QRAC

As we see in Section 2.2, we have to extend the problem Hamiltonian H_{relax} for $(3,2)$-QRAC. Fortunately, we can achieve this step by just substituting the Pauli X, Y, and Z operators that appeared in H_{relax} by the two-qubit operators X', Y', and Z' respectively and changing the coefficient of the 2-local Pauli operators to 1. The definitions of X', Y', and Z' are given in the following equations:

$$X' := \frac{1}{2} X_1 X_2 + \frac{1}{2} X_1 Z_2 + Z_1 I_2, \tag{11}$$

$$Y' := \frac{1}{2} I_1 X_2 + I_1 Z_2 + \frac{1}{2} Y_1 Y_2, \tag{12}$$

$$Z' := Z_1 Z_2 - \frac{1}{2} X_1 I_2 - \frac{1}{2} Z_1 X_2. \tag{13}$$

The algorithms are almost the same as QRAO using $(3,1)$-QRAC. The first step of the algorithm is to color the vertices of the graph. After that, we make pairs of two qubits and assign a single pair to up to

3 vertices for which the same color is assigned in graph coloring. For each vertex assigned to the same pair of two qubits, X', Y', and Z' is assigned in order instead of the Pauli X, Y, and Z operators. Now, all vertices of the graph are associated with one of the operators X', Y', and Z' acting on the same or distinct pair of two qubits. Then, the problem Hamiltonian of the quantum relaxation using $(3,2)$-QRAC denoted by H'_{relax} is defined like the following:

$$H'_{relax} := \frac{1}{2} \sum_{e_{i,j} \in E(G)} (I - P'_i P'_j) \tag{14}$$

where P'_i is one of the operators $\{X', Y', Z'\}$ associated with the vertex v_i. The next step is to find a maximum eigenstate of the relaxed Hamiltonian H'_{relax} by variational methods such as VQE. Once we obtained the quantum states corresponding to the relaxed solution to the MaxCut problem, the quantum state rounding algorithm is performed to extract the classical solution. By using the POVMs in Equations (8) to (10), we can define the rounding algorithm which decodes the encoded bits one by one like the Pauli rounding algorithm of QRAO. On the other hand, to obtain the approximation ratio bound, we need the other rounding algorithm which decodes the configuration of the graph cut by one-shot measurement like the magic state rounding algorithm of QRAO because the Pauli rounding type algorithms do not take the correlation between qubits into account. The key to constructing the rounding algorithm for approximation ratio is to design the quantum measurement which decodes encoded three bits for each qubit at once. We name the algorithm *simultaneous rounding* and define it like the following. In the case of $(3,1)$- or $(2,1)$-QRACs, decoding was performed for each pair of two bit-inverted relationships by using the magic state basis measurements. In the case of $(3,2)$-QRAC, the measurement performed is a two-qubits measurement. There will be up to four different measurement results meaning that up to four different bit patterns can be decoded simultaneously. As we mentioned in Section 3.1, if we know the parity of the encoded bits, then we can decode the encoded three bits by using the 4-outcome quantum measurement defined below up to the parity 0 or 1.

$$\{\rho'_{x_1,x_2,x_3}\}_{x_1 \oplus x_2 \oplus x_3 = 0}, \text{ or } \{\rho'_{x_1,x_2,x_3}\}_{x_1 \oplus x_2 \oplus x_3 = 1}. \tag{15}$$

In the simultaneous rounding algorithm, one of the parity is chosen randomly for each qubit, and one of the corresponding measurements in Equation (15) is performed to the relaxed state ρ'_{relax}. These measurements are performed for all qubits at once and decode one solution to the MaxCut problem. By repeating this procedure sufficient times and taking the best solution, the simultaneous rounding algorithm for the quantum relaxation using $(3,2)$-QRAC finds a classical solution. In this paper, we obtained the approximation ratio bound of the quantum-relaxation based optimizer using $(3,2)$-QRAC for the MaxCut problem under the premise that the found relaxed state's energy is larger than the energy of the quantum state associated with the optimum solution:

Theorem 4 *Consider an oracle ${\mathcal{O}'}_{relax}$ which prepares the relaxed state ρ'_{relax} for the quantum relaxation using $(3,2)$-QRAC satisfying the condition $\mathrm{Tr}[H'_{relax}\rho'_{relax}] \geq OPT$ where OPT is the optimum value. Given access to ${\mathcal{O}'}_{relax}$, the simultaneous rounding algorithm produces a solution to the MaxCut problem with an expected approximation ratio $\mathbb{E}[\gamma] \geq \frac{13}{18} \approx 0.722$.*

Table 1 shows our result for the quantum relaxation using $(3,2)$-QRAC (denoted by $(3,2)$-QRAO) and the previous results by Fuller et al. for QRAOs. Our result is consistent with the trade-off between the bit-to-qubit compression ratio and the approximability of quantum-relaxation based optimizers.

3.3 Space Compression Ratio Preserving Quantum Relaxation

Though QRAO using $(3,1)$- or $(2,1)$-QRACs have a constant-factor space advantage against typical quantum optimizers, the bit-to-qubit compression ratio becomes lower as the density of the graph instance increases. This is because there is a constraint that the endpoints of each edge must be associated with different qubits. In such cases, the quantum-relaxation based optimizer has no space advantage against

Table 1: The relationship between the approximation ratio for the maximum cut problem and the space compression ratio of quantum-relaxation based optimization algorithms

Algorithm	space compression ratio	approximation ratio
$(1,1)$-QRAO [4] ($\approx$ QAOA [3])	1.0	(1.0)
$(2,1)$-QRAO [4]	2.0	0.625
$(3,1)$-QRAO [4]	3.0	0.555
$(3,2)$-QRAO	1.5	0.722 (our result)

(a) $(3,1)$-QRAC **(b)** Encoding of Equation (16)

Figure 3: The Bloch sphere representation of $(3,1)$-QRAC and the encoding of Equation (16)

standard QAOA and VQE algorithms. In this section, we propose new types of encoding which encode up to two classical bits into a single qubit by using $(3,1)$-QRAC. Concretely, we encode the parity of the two bits to the third bit's position in $(3,1)$-QRAC formulation like the following:

$$(x_1, x_2) \mapsto \tilde{\rho}_{x_1,x_2} := \frac{1}{2}\left(I + \frac{1}{\sqrt{3}}((-1)^{x_1}X + (-1)^{x_2}Y + (-1)^{x_1 \oplus x_2}Z)\right). \tag{16}$$

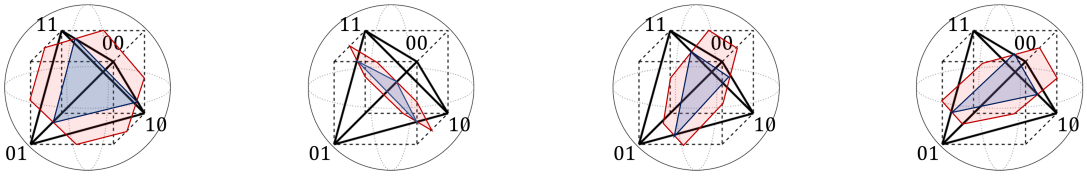
Figure 3 shows the Bloch sphere representation of $(3,1)$-QRAC and the encoding of Equation (16). Equation (16) encodes the two classical bits into one of the four vertices of the tetrahedron visualized in Figure 3b. These four vertices correspond to the four of eight vertices of the cube in the case of $(3,1)$-QRAC in Figure 3a. Let us formulate the quantum relaxation based on the encoding in Equation (16). In the QRAO by Fuller et al., a graph coloring algorithm is performed as preprocessing to satisfy the constraint that the endpoints of each edge must be assigned to different qubits. On the contrary, in our new space compression ratio preserving quantum relaxation, such preprocessing is unnecessary. We just partition the vertices into $\frac{|V(G)|}{2}$ pairs of two vertices and assign the Pauli X or Y to the two vertices respectively. Then, we construct the relaxed Hamiltonian from the instance graph. For each edge $(i,j) \in E(G)$, if the endpoints of it are assigned to different qubits, we encode the edge as the term P_iP_j where $P_i \in \{X, Y\}$ are the Pauli operators associated with the vertex of index i. If the endpoints of the edge are assigned to the same qubit, we use the Pauli Z operator acting on the qubit. Let $\mathrm{Q\_idx}(i)$ be the index of the qubit associated with the i-th vertex. Formally, the relaxed Hamiltonian for our quantum relaxation $\tilde{H}_{relax}$ is defined like the following:

$$\tilde{H}_{relax} := \frac{1}{2}\sum_{e:=(i,j)\in E(G)} (I - O_e) \tag{17}$$

where

$$O_e := \begin{cases} 3P_iP_j & \text{if } \mathrm{Q\_idx}(i) \neq \mathrm{Q\_idx}(j), \\ \sqrt{3}Z_k & \text{if } \mathrm{Q\_idx}(i) = \mathrm{Q\_idx}(j) = k. \end{cases} \tag{18}$$

We note that P_i and P_j in Equation (18) are X or Y acting on the different qubits $\mathrm{Q\_idx}(i)$ and $\mathrm{Q\_idx}(j)$. As well as the other quantum relaxations, we explore the maximum eigenstate of $\tilde{H}_{relax}$ and find the candidate relaxed state $\tilde{\rho}_{relax}$. The next step is to define the quantum state rounding algorithm. The Pauli rounding is the same as that for $(3,1)$-QRAO but disregards the third encoded bit. The magic state



(a) $\mu_1^{\pm}$ (b) $\mu_2^{\pm}$ (c) $\mu_3^{\pm}$ (d) $\mu_4^{\pm}$

Figure 4: The intuition of the quantum measurements performed in magic state rounding algorithm performed in space compression ratio preserving quantum relaxation

rounding algorithm for our quantum relaxation is also the same as that for $(3,1)$-QRAO but the decoding rule is different. By the magic bases $\mu_1^{\pm}, \mu_2^{\pm}, \mu_3^{\pm}, \mu_4^{\pm}$, the four encoded patterns $00, 01, 10, 11$ are divided into 2 groups containing 1 and 3 patterns. The intuition of the magic state measurement is described in Figure 4. For instance, $\mu_1^{\pm}$ divides the patterns into $\{00\}$ and $\{01, 10, 11\}$. If the measurement outcome of $\mu_1^{\pm}$ is 0, then the encoded bits are decided to be 00. Otherwise, the probabilities that the encoded bits are 01, 10, and 11 are the same $\left(\frac{1}{3}\right)$. From the above discussions, we define the decoding rule for $\mu_1^{\pm}$ as

$$0 \mapsto 00,$$
$$1 \mapsto 01 \text{ or } 10 \text{ or } 11 \text{ with the same probabilities.}$$

We define the decoding rules in the same way for $\mu_2^{\pm}, \mu_3^{\pm}, \mu_4^{\pm}$. Our interest is the approximation ratio bound of our space compression ratio preserving quantum relaxation. Unfortunately, we didn't obtain the constant expected approximation ratio for it. Instead, we have the approximation ratio bound dependent on the ratio of the edges whose endpoints are associated with different qubit denoted by $\lambda \in [0,1]$ and the parameter $\epsilon \in \left[0, \frac{1}{2}\right]$ defined by the equation: $OPT = \left(\frac{1}{2} + \epsilon\right)|E(G)|$. We note that ϵ is called the gain, and the problem to calculate the value ϵ is called MaxCutGain [2].

Theorem 5 *Let $\lambda \in [0,1]$ be the ratio of the edges whose endpoints are associated with different qubits. Let $\epsilon \in [0, \frac{1}{2}]$ be the gain. Consider an oracle $\tilde{\mathcal{O}}_{relax}$ which prepares the relaxed state for the space compression ratio preserving quantum relaxation using the encoding in Equation* (16) *satisfying the condition $\mathrm{Tr}[\tilde{H}_{relax}\tilde{\rho}_{relax}] \geq OPT$ where OPT is the optimum value. Given access to $\tilde{\mathcal{O}}_{relax}$, the magic state rounding algorithm defined in this section produces a solution to the MaxCut problem with an expected approximation ratio*

$$\mathbb{E}[\gamma] \geq \max\left\{\frac{3 - 2\lambda + 2\epsilon}{3 + 6\epsilon}, \frac{9 - 2\sqrt{3} + 2\sqrt{3}\lambda + 2\epsilon}{9 + 18\epsilon}\right\}.$$

Consider the condition of λ and ϵ when our quantum relaxation has non-obvious approximation ratio.

$$\mathbb{E}[\gamma] > \frac{1}{2} \iff \begin{cases} 0 \leq \lambda \leq 1 & \text{if } \epsilon < \frac{9-\sqrt{3}}{14+2\sqrt{3}} \approx 0.416 \\ 0 \leq \lambda < \frac{3-2\epsilon}{4}, -\frac{9}{4\sqrt{3}} + 1 + \frac{7}{2\sqrt{3}}\epsilon < \lambda \leq 1 & \text{if } 0.416 \approx \frac{9-\sqrt{3}}{14+2\sqrt{3}} \leq \epsilon \leq \frac{1}{2} \end{cases} \tag{19}$$

If the graph instance has a relatively small MaxCut value (i.e. the gain $\epsilon < 0.416$), the space compression ratio preserving quantum relaxation has a non-trivial approximation ratio bound for arbitrary lambda. It means that we do not have to care about anything when assigning vertices to the qubits in the preprocessing.

4 Future Directions

We consider the information-theoretic analysis of the trade-off between the approximation ratio and the space compression ratio of the quantum relaxation, which seems to contribute to revealing the theoretical

limitation of the quantum-relaxation based approaches. From the result of QRAO [4] and our result of the quantum relaxation using $(3, 2)$-QRAC, we conjectured the approximation ratio of the quantum-relaxation using a QRAC with the bit-to-qubit compression ratio r as $\frac{1}{2}\left(1 + r^{-2}\right)$. Our space compression ratio preserving quantum relaxation is not included in the quantum relaxations mentioned in the above conjecture because it does not use the formulation of $(3, 1)$-QRAC directly. The difficulty of the proof of this conjecture lies in the point that the concrete formulations of QRACs for general m and n are not known. The $(3, 2)$-QRAC is obtained by numerical calculations, and it is hard to extend the rule of the construction of the QRAC to general m and n. By combining the results of the approximation ratio for the problem to find a maximum eigenstate of the Hamiltonians with the above conjecture, the quantum approximability without assumptions for the MaxCut problem can be obtained.

References

[1] Andris Ambainis, Ashwin Nayak, Amnon Ta-Shma, and Umesh Vazirani. Dense quantum coding and quantum finite automata. *Journal of the ACM (JACM)*, 49(4):496–511, 2002.

[2] M. Charikar and A. Wirth. Maximizing quadratic programs: extending Grothendieck's inequality. In *45th Annual IEEE Symposium on Foundations of Computer Science*, pages 54–60, 2004.

[3] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm. *arXiv preprint arXiv:1411.4028*, 2014.

[4] Bryce Fuller, Charles Hadfield, Jennifer R Glick, Takashi Imamichi, Toshinari Itoko, Richard J Thompson, Yang Jiao, Marna M Kagele, Adriana W Blom-Schieber, Rudy Raymond, et al. Approximate solutions of combinatorial problems via quantum relaxations. *arXiv preprint arXiv:2111.03167*, 2021.

[5] Michel X Goemans and David P Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM (JACM)*, 42(6):1115–1145, 1995.

[6] Masahito Hayashi, Kazuo Iwama, Harumichi Nishimura, Rudy Raymond, and Shigeru Yamashita. (4, 1)-quantum random access coding does not exist—one qubit is not enough to recover one of four bits. *New Journal of Physics*, 8(8):129, 2006.

[7] Alexander Semenovich Holevo. On capacity of a quantum communications channel. *Problemy Peredachi Informatsii*, 15(4):3–11, 1979.

[8] Takashi Imamichi and Rudy Raymond. Constructions of quantum random access codes. In *Asian Quantum Information Symposium (AQIS)*, volume 66, 2018.

[9] Julia Kempe, Alexei Kitaev, and Oded Regev. The complexity of the local Hamiltonian problem. *SIAM Journal on Computing*, 35(5):1070–1097, 2006.

[10] Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O'Donnell. Optimal inapproximability results for MAX-CUT and other 2-variable CSPs? *SIAM Journal on Computing*, 37(1):319–357, 2007.

[11] Laura Mančinska and Sigurd AL Storgaard. The geometry of Bloch space in the context of quantum random access codes. *Quantum Information Processing*, 21(4):1–16, 2022.

[12] Giacomo Nannicini. Performance of hybrid quantum-classical variational heuristics for combinatorial optimization. *Physical Review E*, 99(1):013304, 2019.

[13] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J Love, Alán Aspuru-Guzik, and Jeremy L O'brien. A variational eigenvalue solver on a photonic quantum processor. *Nature communications*, 5(1):1–7, 2014.

[14] Kosei Teramoto, Rudy Raymond, and Hiroshi Imai. The role of entanglement in quantum-relaxation based optimization algorithms. *arXiv preprint arXiv:2302.00429*, 2023.

[15] Kosei Teramoto, Rudy Raymond, Eyuri Wakakuwa, and Hiroshi Imai. Quantum-relaxation based optimization algorithms: Theoretical extensions. *arXiv preprint arXiv:2302.09481*, 2023.

[16] Dominic JA Welsh and Martin B Powell. An upper bound for the chromatic number of a graph and its application to timetabling problems. *The Computer Journal*, 10(1):85–86, 1967.

[17] Andrew Zhao and Nicholas C. Rubin. Quantum relaxation for quadratic programs over orthogonal matrices, 2023.

Absence of percolation in graphs based on stationary point processes with degrees bounded by two

Benedikt Jahnel

Weierstrass Institute Berlin
Mohrenstraße 39, 10117 Berlin, Germany,
and Technische Universität Braunschweig,
Institute of Mathematical Stochastics,
Universitätsplatz 2, 38106 Braunschweig,
Germany
jahnel@wias-berlin.de

András Tóbiás

Department of Computer Science and
Information Theory
Budapest University of Technology and
Economics
Műegyetem rakpart 11., 1111 Budapest,
Hungary,
and Alfréd Rényi Institute of Mathematics,
Reáltanoda utca 13–15., 1053 Budapest,
Hungary
tobias@cs.bme.hu

Abstract: We consider the bidirectional k-nearest neighbor graph based on a stationary point process, where one connects two points of the point process by an edge if and only if they are mutually among the k nearest neighbors of each other. For a large class of stationary point processes in arbitrary dimensions, we show that for $k = 2$, the arising graph has no infinite connected component, almost surely. In the particular case of the two-dimensional homogeneous Poisson point process, this verifies a conjecture by Balister and Bollobás.

Keywords: Continuum percolation, stationary point processes, degree bounds, bidirectional k-nearest neighbor graph, edge-preserving property, deletion-tolerance.

1 Introduction

This entire document is an extended abstract of the paper [10]. For all the proofs and discussions that we are omitting here, as well as for the acknowledgements and the references not mentioned in this abstract, see the full paper.

Continuum percolation was introduced by Gilbert [7] in order to model connectivity in large telecommunication networks. In his graph model, the vertices form a homogeneous Poisson point process (PPP) of (spatial) intensity $\lambda > 0$ in $\mathbb{R}^2$, that is, the number of points in a measurable subset of $\mathbb{R}^2$ is Poisson distributed with parameter equal to λ times the Lebesgue measure of the subset, and the numbers of points in disjoint subsets are independent. Now, two points are connected by an edge whenever their distance is less than a fixed connection radius $r > 0$. He showed that this model undergoes a phase transition: λ is sufficiently small, then the graph consists of finite components only, almost surely, whereas for large enough λ, the graph *percolates*, i.e., it has an unbounded connected component, also almost surely.

This model has been widely extended, for instance to the case of random connection radii and for various point processes (see [10, Section 1] for further references). The homogeneous Poisson point process in $\mathbb{R}^2$ (or in $\mathbb{R}^d$ in general, defined analogously) represents a fully random set of points with total independence among the numbers of points in disjoint subsets. This independence property in fact already implies that the number of points in a given measurable subset of $\mathbb{R}^d$ has to be Poisson distributed, see [K93, Section 1.4]. The homogeneous PPP is a natural toy model for the set of users in a spatial telecommunication network. On the other hand, it does not capture e.g. spatial inhomogeneities due to geographic reasons, correlations among the numbers of users in different areas, or the fact that users are mainly situated along streets, but other point processes can be chosen in order to make the

modelling more realistic. Our basic assumption will be that the point process under consideration is stationary, i.e., its distribution is shift-invariant.

A drawback of Gilbert's model is that it allows for an arbitrarily large degree of the vertices, whereas for many applications, it is a reasonable assumption that the vertices should have bounded degree. Incorporating this property, Häggström and Meester [8] studied percolation in the so-called *undirected k-nearest neighbor* ($\mathbf{U}$-kNN) *graph*, based on a stationary PPP in $\mathbb{R}^d$, $d \geq 1$. Here, all points of the point process are connected to their k-nearest neighbors, for some fixed $k \in \mathbb{N}$. This results in a graph that is the undirected variant of a directed graph with out-degrees bounded by k, which itself also has degrees larger than k. Let us write $k_{\mathbf{U},d}$ for the minimum of all $k \in \mathbb{N}$ such that the $\mathbf{U}$-kNN-graph of the stationary PPP in $\mathbb{R}^d$ percolates with positive probability. It was shown in [8] that $k_{\mathbf{U},d} > 1$ for all $d \in \mathbb{N}$, however, $k_{\mathbf{U},d} < \infty$ for all $d \geq 2$ and $k_{\mathbf{U},d} = 2$ for all sufficiently large d.

Balister and Bollobás [1] studied the case $d = 2$. They also introduced another undirected graph, which is contained in the $\mathbf{U}$-kNN graph, called the *bidirectional k-nearest neighbor* ($\mathbf{B}$-kNN) *graph*. Here, one connects two points of the point process if and only if they are mutually among the k nearest neighbors of each other. This graph has in fact degrees bounded by k, which immediately implies that there is no percolation for $k = 1$, whatever the vertex set is. The critical out-degree $k_{\mathbf{B},d}$ is defined analogously to $k_{\mathbf{U},d}$ but with $\mathbf{U}$ replaced by $\mathbf{B}$. It was shown in [1] that $k_{\mathbf{U},2} \leq 11$ and $k_{\mathbf{B},2} \leq 15$. Further, 'high-confidence results' of [1] indicate $k_{\mathbf{U},2} = 3$ and $k_{\mathbf{B},2} = 5$. By 'high-confidence results', the authors of that paper meant that these assertions follow once one shows that a certain deterministic integral exceeds a certain deterministic value, however, the integrals were only evaluated via Monte–Carlo methods so far. Hence, from a mathematical point of view, these are still open conjectures, but there is very strong numerical evidence that they are true (see Figure 1 below for an illustration).

In the present work, we focus on the $\mathbf{B}$-kNN graph in arbitrary dimension for $k = 2$ and we verify that it does not percolate under the general assumption that the underlying stationary point process is *deletion-tolerant* in the sense of [9]. In general, if in an undirected graph all degrees are bounded by $k = 2$, all infinite connected components must be path graphs (no cycles, no branchings), infinite in one or two directions. This makes the graph similar to a one-dimensional continuum percolation model, indicating that, under rather general conditions, there should be no infinite connected component. Certainly, there are deterministic point processes where percolation is possible, but a little bit of randomness can be expected to suffice for non-percolation. In our recent paper [11], we showed that in so-called *signal-to-interference ratio* (SINR) graphs based on general stationary Cox point processes in any dimension, under rather general choices of the parameters resulting in degrees bounded by 2, there is no percolation. SINR graphs are a popular model for modelling connectivity in wireless networks [4, 5, 13, 11], being proper subgraphs of the $\mathbf{B}$-kNN graph (in certain cases, this is only true with a slight modification of the definition of the $\mathbf{B}$-kNN graph). Hence, the lack of percolation in the SINR graph does not imply the same in the $\mathbf{B}$-kNN graph. Nevertheless, the proof of absence of percolation in SINR graphs with degrees bounded by 2 can be extended in order to disprove percolation in the $\mathbf{B}$-2NN graph, even for general deletion-tolerant stationary point processes, and this includes the case of stationary Cox point processes and in particular also of the homogeneous PPP.

Thus, our results imply that $k_{\mathbf{B},2} \geq 3$, which provides a partial verification of the high-confidence results of [1]. Let us also note that in [10] we do not only verify the absence of percolation for the $\mathbf{B}$-kNN graph, but for a generalization called the f-kNN graph. This makes it possible to consider an arbitrary norm on $\mathbb{R}^d$ instead of the Euclidean norm, and the notion of being k-nearest neighbors in the graph does not only depend on distances (w.r.t. this norm) but also possibly on random marks associated with the points of the point process via a function f. In particular, any SINR graph is a proper subgraph of an f-kNN graph. This general setting hardly requires changes in the proofs, and hence we omit it from the present abstract for brevity.

2 Model definition and main result

Our setting is as follows. Let $d \in \mathbb{N}$, and let $\|\cdot\|$ be the Euclidean norm on $\mathbb{R}^d$. Next, let $X = \{X_i\}_{i\in I}$ be a a stationary point process in $\mathbb{R}^d$ with finite intensity $\lambda = \mathbb{E}[X([0,1]^d)]$, that is *nonequidistant*. This means that for all $i, j, k, l \in I$, $\|X_i - X_j\| = \|X_k - X_l\| > 0$ implies $\{i, j\} = \{k, l\}$ and $\|X_i\| = \|X_j\|$ implies $i = j$, almost surely. Clearly, this property implies that the point process X is *simple*, i.e., $\mathbb{P}(X_i \neq X_j, \forall i, j \in I \text{ with } i \neq j) = 1$. For illustration, note that the randomly shifted lattice $\mathbb{Z}^d + U$, where U is a uniform random variable in $[0,1]^d$, is a simple, stationary, but not nonequidistant point process on $\mathbb{R}^d$.

Thus, if $x = \{x_i\}_{i\in I}$ is a deterministic, locally finite, infinite and nonequidistant set of points in $\mathbb{R}^d$ (for some countable index set I) and $v_o \in x$, we can represent x as $x = \{\mathrm{v}_n(v_o, x)\}_{n\in\mathbb{N}_0}$, where $\mathrm{v}_0(v_o, x) = v_o$, and $\mathrm{v}_n(v_o, x)$ is the n-th nearest neighbor of v_o in x with respect to the Euclidean distance for any $n \in \mathbb{N}_0$. As already indicated, for $k \in \mathbb{N}$, the *bidirectional k-nearest neighbor* (**B**-*kNN*) *graph* is the random undirected graph $g_{\mathbf{B},k}(X)$ having vertex set X and for all $i \in I$ and $n \in \{1, \ldots, k\}$ an edge between X_i and $\mathrm{v}_n(X_i, X)$ whenever $X_i \in \{\mathrm{v}_1(\mathrm{v}_n(X_i, X), X), \ldots, \mathrm{v}_k(\mathrm{v}_n(X_i, X), X)\}$ (see Figure 1).

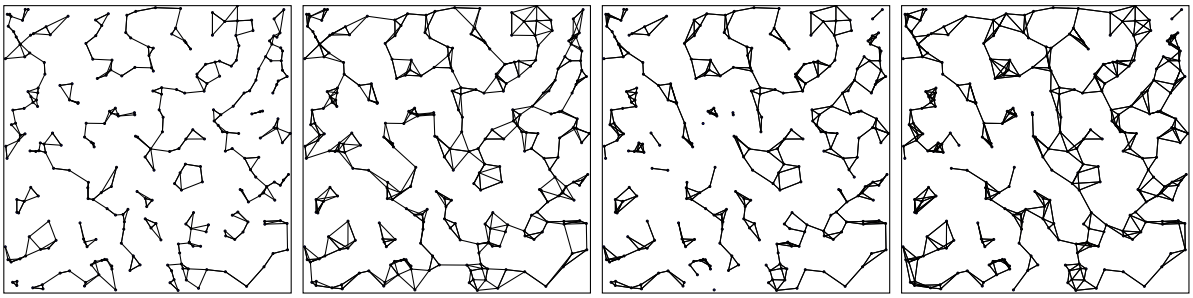


Figure 1: Different kNN graphs based on the same realization of a two-dimensional homogeneous PPP. From left to right: **U**-kNN graphs for $k = 2, 3$ and **B**-kNN graphs for $k = 4, 5$.

Apart from the basic requirement of being nonequidistant, the property of *deletion-tolerance* introduced in [9] is the most important requirement on the point process. An *X-point* is an $\mathbb{R}^d$-valued random variable Z, defined on the same probability space as X, such that $Z \in X$ a.s., and one says that X is *deletion-tolerant* if for any X-point Z, the distribution of $X \setminus \{Z\}$ is absolutely continuous with respect to the one of X. See [9, Theorem 1.1] for equivalent formulations of this property. Equipped with the above definitions, we are now able to state our main result.

Theorem 1 *Let the deletion-tolerant point process X be stationary, nonequidistant and of a finite intensity. Then, $\mathbb{P}(g_{\mathbf{B},2}(X)$ percolates$) = 0$.*

The proof of this theorem is sketched in Section 3. In the rest of this section, let us mention some examples, counterexamples and consequences, whose proofs can only be found in [10]. Immediate extensions of our method require that the stationary undirected graph with degrees bounded by 2 is *edge-preserving*, which is a property that we will explain in the sketch of the proof of the theorem below. This way, the proof can be applied, e.g., to the graph where we connect each vertex to its two *furthest* neighbors in a bidirectional sense within a ball of fixed radius around the vertex (whenever they exist), but not to the graph where we connect each vertex to its k_1-th and k_2-th nearest neighbours in a bidirectional sense if $\{k_1, k_2\} \neq \{1, 2\}$.

Note that many point processes are deletion-tolerant, e.g., all stationary Cox point processes (and thus in particular all homogeneous PPPs) and a large class of Gibbs point processes (see [10, Proposition 2.6]). The class of deletion-tolerant Gibbs point processes is very rich and in particular includes the classical examples of superstable Hamiltonians, see [3] and references therein. Moreover, let us mention some well-known point processes that are not deletion-tolerant. As introduced in [6, 12], we say that

the point process X is *number rigid* if for any $\mathcal{K} \subset \mathbb{R}^d$ compact, there exists a deterministic measurable function $h_{\mathcal{K}}$ such that,

$$\#(X \cap \mathcal{K}) = h_{\mathcal{K}}(X \setminus \mathcal{K}),$$

almost surely, i.e., X outside $\mathcal{K}$ determines the number of points of X in $\mathcal{K}$.

Proposition 2 *If the point process X is stationary and number rigid with positive intensity, then X is not deletion-tolerant.*

This proposition follows immediately from results of [9]. Important examples of point processes in $\mathbb{R}^2$ satisfying the condition of Proposition 2 are the Ginibre example and the Gaussian zero process, see [6]. Although the proof of Theorem 1 is not applicable for these point processes, we conjecture that they satisfy the assertion of the theorem.

3 Sketch of proof of Theorem 1

A *cluster* in $g_{\mathbf{B},2}(X)$ is a maximal connected component. The proof of Theorem 1 proceeds along the following line of arguments.

Step 1: We show that with probability 1, $g_{\mathbf{B},2}(X)$ contains no degree-1 point included in an infinite cluster.
Main tool: Mass-transport principle.

This claim was already proven in our previous paper (see [11, Lemma 5.4]), and in fact it holds for any stationary random graph based on a nonequidistant point process X with degrees bounded by 2. The proof is a certain variant of the *mass-transport principle* (see [2, Section 4.2] for instance). Vaguely speaking, the argument is the following. By stationarity, if there exists a degree-1 point in an infinite cluster, then such points must have a positive density. However, since degrees are bounded by 2, any infinite cluster contains at most one degree-1 point and infinitely many degree-2 points, which implies that the aforementioned density must be zero.

Step 2: Conditioning on having an infinite cluster, removing a finite set of points and obtaining a set of points whose **B**-2NN graph has a degree-1 point included in an infinite cluster.
Main tool: Edge-preserving property.

For the rest of the proof, we assume for a contradiction that $g_{\mathbf{B},2}(X)$ percolates with positive probability. Then, by Step 1, there also exists an infinite cluster consisting only of degree-2 points with positive probability. Conditional on the latter event, let $z(X)$ denote the closest point of X to the origin that is contained in such an infinite cluster. Then, by construction, $z(X)$ is connected to both of its two nearest neighbors $\mathrm{v}_1(z(X), X)$ and $\mathrm{v}_2(z(X), X)$ by an edge. Let $\tau(X)$ denote the (random) smallest number $i \geq 3$ such that $\mathrm{v}_{\tau(X)}(z(X), X)$ is included in the infinite cluster of $z(X)$ (necessarily, there exists such i because the cluster containing $z(X)$ is infinite). Now, at least one of the two nearest neighbors $\mathrm{v}_1(z(X), X)$ and $\mathrm{v}_2(z(X), X)$ of $z(X)$ in X is not connected to $\mathrm{v}_{\tau(X)}(z(X), X)$ by an edge because otherwise we would obtain a cycle in the infinite cluster containing $z(X)$, which is impossible because degrees are bounded by 2. Let us denote this neighbor by $m(X)$; if none of $\mathrm{v}_1(z(X), X)$ and $\mathrm{v}_2(z(X), X)$ are connected to $\mathrm{v}_{\tau(X)}(z(X), X)$ by an edge, let us put $m(X) = \mathrm{v}_1(z(X), X)$. Let $n(X)$ denote the element of $\{\mathrm{v}_1(z(X), X), \mathrm{v}_2(z(X), X)\}$ unequal to $m(X)$.

A key property of the **B**-kNN graph is that it is *edge-preserving*. That is, after removing a subset of vertices and redrawing the graph based on the remaining vertices according to the same rules, edges between remaining vertices are preserved. Hence, if we now remove $Y := \{m(X), \mathrm{v}_3(X), \ldots, \mathrm{v}_{\tau(X)-1}(X)\}$ from the vertex set X, in the **B**-2NN graph $g_{\mathbf{B},2}(X \setminus Y)$ of the remaining vertex set, $z(X)$ is still included in an infinite cluster. This is true because we have only removed the edge between $z(X)$ and $m(X)$ from the cluster, so that the infinite path starting from $z(X)$ with the edge towards $n(X)$ is still completely preserved. However, $z(X)$ has degree 1 in this cluster. Indeed, by construction, $z(X)$ can only be connected by an edge to its two nearest neighbors in the new graph $g_{\mathbf{B},2}(X \setminus Y)$. The first nearest neighbor of $z(X)$ is now $n(X)$ since $m(X)$ has been removed from the graph, and the second $z(X)$ is

now $\mathrm{v}_{\tau(X)}(z(X), X)$ because Y has been removed. However, since $\mathrm{v}_{\tau(X)}(z(X), X)$ originally had degree 2 within the infinite cluster of $z(X)$ and none of these neighbors was equal to an element of Y, in $g_{\mathbf{B},2}(X \setminus Y)$ it is still connected to the same two vertices, none of which is equal to $z(X)$, and thus it has no free degrees left that $z(X)$ could use. We encourage the reader to consult [10, Figure 2] for an illustration of the simplest case $\tau(X) = 3$.

We conclude that after removing the aforementioned vertices, in $g_{\mathbf{B},2}(X \setminus Y)$, $z(X)$ is in an infinite cluster and has degree 1. Extend the definition of Y via putting $Y = \varnothing$ on the event that there exists no infinite cluster in $g_{\mathbf{B},2}(X)$ or all infinite clusters contain a degree-1 point.

Step 3: Obtaining a contradiction.
Main tool: An equivalent characterization of deletion-tolerance via finite subprocesses.

According to Step 1, starting from realizations of $g_{\mathbf{B},2}(X)$ having an infinite cluster consisting only of degree-2 points, via the procedure described in Step 2 we obtain realizations of $X \setminus Y$ that are altogether included in a set of probability zero *with respect to the distribution of* $g_{\mathbf{B},2}(X)$. We now want to conclude that this implies that the initial realizations (and thus also the set of all realizations of $g_{\mathbf{B},2}(X)$ having an infinite cluster) are included in a nullset w.r.t. the same distribution. But (using Step 1 once more) this contradicts the assumption that $g_{\mathbf{B},2}(X)$ has an infinite cluster with positive probability, which finishes the proof of the theorem.

This is the part of the proof where deletion-tolerance plays an important role. We say that a point process Z is a *finite subprocess* of X if Z is defined on the same probability space as X, satisfies $\#Z < \infty$ and $Z \subset X$ almost surely. Then it was shown in [9, Theorem 1.1] that if X is deletion-tolerant, then, for any finite subprocess Z of X, the law of $X \setminus Z$ is absolutely continuous with respect to the one of X. For example, Y is a finite subprocess of X.

Using this assertion and the result of Step 2, it is straightforward derive the aforementioned contradiction. However, the notation needed for a full proof is somewhat extensive, and therefore we refrain from presenting further details in this abstract; see [10, Section 3].

References

[1] P. Balister and B. Bollobás, Percolation in the k-nearest neighbor graph, In *Recent Results in Designs and Graphs: a Tribute to Lucia Gionfriddo, Quaderni di Matematica*, **28**, Editors: M. Buratti and C. Lindner and F. Mazzocca and N. Melone (2013)

[2] D. Coupier, D. Dereudre and S. Le Stum, Absence of percolation for Poisson outdegree-one graphs, *Ann. Inst. Henri Poincaré Probab. Stat.*, **56:2** (2020)

[3] D. Dereudre, Introduction to the theory of Gibbs point processes, In *Stochastic Geometry*, Springer (2019)

[4] O. Dousse, F. Baccelli and P. Thiran, Impact of interferences on connectivity in ad hoc networks, *IEEE/ACM Trans. Networking*, **1** (2005)

[5] O. Dousse, M. Franceschetti, N. Macris, R. Meester, and P. Thiran, Percolation in the signal to interference ratio graph, *J. Appl. Probab.*, **43** (2006)

[6] S. Ghosh and Y. Peres, Rigidity and tolerance in point processes: Gaussian zeroes and Ginibre eigenvalues, *Duke Math. J.*, **166:10** (2017)

[7] E.N. Gilbert, Random plane networks, *J. Soc. Indust. Appl. Math.*, **9** (1961)

[8] O. Häggström and R. Meester, Nearest neighbor and hard sphere models in continuum percolation, *Random Structures and Algorithms*, **9** (1996)

[9] A.E. Holroyd and T. Soo, Insertion and deletion tolerance of point processes, *Electron. J. Probab.*, **18**, paper no. 74 (2013)

[10] B. Jahnel and A. Tóbiás, Absence of percolation in graphs based on stationary point processes with degrees bounded by two, *Random Structures and Algorithms*, **62:1** (2022)

[11] B. Jahnel and A. Tóbiás, SINR percolation for Cox point processes with random powers, *Adv. Appl. Probab.*, **54:1** (2022)

[K93] J.F.C. Kingman, *Poisson Processes*, Oxford University Press, New York (1993)

[12] M.A. Klatt and G. Last, On strongly rigid hyperfluctuating random measures, *J. Appl. Probab.*, **59:4** (2022)

[13] A. Tóbiás, Signal to interference ratio percolation for Cox point processes, *Lat. Am. J. Probab. Math. Stat.*, **17** (2020)

Geodesic Diameter in Polygons with Holes

Adrian Dumitrescu

Algoresearch L.L.C.
Milwaukee, WI 53217, USA
ad.dumitrescu@algoresearch.org

Csaba D. Tóth

Department of Mathematics
California State University Northridge
Los Angeles, CA 91330, USA
csaba.toth@csun.edu

Abstract: For a polygon P with holes in the plane, we denote by $\varrho(P)$ the ratio between the geodesic and the Euclidean diameters of P. It is shown that over all convex polygons with h convex holes, the supremum of $\varrho(P)$ is between $\Omega(h^{1/3})$ and $O(h^{1/2})$. However, if all holes are *fat* convex polygons, then $\varrho(P) = O(1)$.

Keywords: combinatorial geometry, diameter, distortion, escape path, polygon with holes

1 Introduction

Determining the maximum distortion between two metrics on the same ground sets is a fundamental problem in metric geometry. In this paper, we study the maximum ratio between the geodesic (i.e., shortest path) diameter and the Euclidean diameter over polygons with holes. A *polygon P with h holes* (also known as a *polygonal domain*) is defined as follows. Let P_0 be a simple polygon, and let $P_1, \ldots, P_h$ be pairwise disjoint simple polygons in the interior of P_0. Then $P = P_0 \setminus \left(\bigcup_{i=1}^{h} P_i\right)$.

For any two points $s, t \in P$, the Euclidean distance is $|st| = \|s - t\|_2$, and the shortest path distance $\mathrm{geod}(s, t)$ is the Euclidean length of the shortest polygonal path between s and t contained in P. The *Euclidean diameter* of P is $\mathrm{diam}_2(P) = \sup_{s,t\in P} |st|$ and its *geodesic diameter* is $\mathrm{diam}_g(P) = \sup_{s,t\in P} \mathrm{geod}(s, t)$. By definition, we have $|st| \le \mathrm{geod}(s, t)$ for any two points $s, t \in P$, hence $\mathrm{diam}_2(P) \le \mathrm{diam}_g(P)$. We are interested in the distortion

$$\varrho(P) = \frac{\mathrm{diam}_g(P)}{\mathrm{diam}_2(P)}.$$

Note that, without further restrictions, $\varrho(P)$ is unbounded, even for simple polygons. Indeed, if P is a zigzag polygon with n vertices, lying in a disk of unit diameter, then $\mathrm{diam}_2(P) \le 1$ and $\mathrm{diam}_g(P) = \Omega(n)$, hence $\varrho(P) \ge \Omega(n)$. It is not difficult to see that this bound is the best possible.

In this paper, we consider convex polygons with convex holes. Specifically, let $\mathcal{C}(h)$ denote the family of polygonal domains $P = P_0 \setminus \left(\bigcup_{i=1}^{h} P_i\right)$, where $P_0, P_1, \ldots, P_h$ are convex polygons; and let

$$g(h) = \sup_{P\in\mathcal{C}(h)} \varrho(P)$$

It is clear that if $h = 0$, then $\mathrm{geod}(s, t) = |st|$ for all $s, t \in P$, which implies $g(0) = 1$. Or main result is the following.

Theorem 1 *For every $h \in \mathbb{N}$, we have $\Omega(h^{1/3}) \le g(h) \le O(h^{1/2})$.*

However, if we further restrict the holes to be *fat* convex polygons, we can show that $\varrho(h)$ is bounded by a constant for all $h \in \mathbb{N}$. In fact for every $s, t \in P$, the distortion $\mathrm{geod}(s, t)/|st|$ is also bounded by a constant.

Informally, a convex body is *fat* if its width is comparable with its diameter. The *width* a convex body C is the minimum width of a slab bounded by parallel lines enclosing C. For $0 \leq \lambda \leq 1$, a convex body C is λ-*fat* if the ratio of its width to its diameter is at least λ, that is, $\text{width}(C)/\text{diam}_2(C) \geq \lambda$; and C is *fat* if the inequality holds for a constant λ. For instance, a disk is 1-fat, a 3×4 rectangle is $\frac{3}{5}$-fat and a line segment is 0-fat. Let $\mathcal{F}_\lambda(h)$ be the family of polygonal domain $P = P_0 \setminus \left(\bigcup_{i=1}^{h} P_i\right)$, where $P_0, P_1, \ldots, P_h$ are λ-fat convex polygons; and let $\mathcal{F}_\lambda = \bigcup_{h=0}^{\infty} \mathcal{F}_\lambda(h)$.

Proposition 2 *For every $P \in \mathcal{F}_\lambda$, we have $\varrho(P) \leq O(\lambda^{-1})$.*

Related work. The geodesic distance in polygons with or without holes have been studied extensively from the algorithmic perspective; see [14] for a comprehensive survey. In a simple polygon P with n vertices, one can compute the geodesic distance between two given points in $O(n)$ time [12], trade-offs are also available between time and workspace [8]. A shortest-path data structure can report the geodesic distance between any two query points in $O(\log n)$ time after $O(n)$ preprocessing time [7]. In $O(n)$ time, one can also compute the geodesic diameter [9] and radius [1].

For polygons with holes, much more involved techniques are needed. Let P be a polygon with h holes, and a total of n vertices. For any $s, t \in P$, one can compute $\text{geod}(s, t)$ in $O(n + h \log h)$ time and $O(n)$ space [17], improving earlier bounds in [10, 11, 13, 18]. A shortest-path data structure can report the geodesic distance between two query points in $O(\log n)$ query time using $O(n^{11})$ space; or in $O(h \log n)$ query time with $O(n + h^5)$ space [4]. The geodesic radius can be computed in $O(n^{11} \log n)$ time [3, 16], and the geodesic diameter in $O(n^{7.73})$ or $O(n^7(\log n + h))$ time [2]. One can find an $(1+\varepsilon)$-approximation in $O((n/\varepsilon^2 + n^2/\varepsilon) \log n)$ time [2, 3]. The geodesic diameter may be attained by a point pair $s, t \in P$, where both s and t lie in the interior or P; in which case it is known [2] that there are at least five different geodesic paths between s and t.

2 Convex Polygons with Convex Holes

In section, we prove Theorem 1. The upper bound is established in Lemma 3 and and a lower bound construction is presented in Lemma 5.

Upper Bound. Let $P \in \mathcal{C}(h)$ for some $h \in \mathbb{N}$ and let $s \in P$. For every hole P_i, let ℓ_i and r_i be points on the boundary of P_i such that $\overrightarrow{s\ell_i}$ and $\overrightarrow{sr_i}$ are tangent to P_i, and P_i lies on the left (resp., right) side of the ray $\overrightarrow{s\ell_i}$ (resp., $\overrightarrow{sr_i}$).

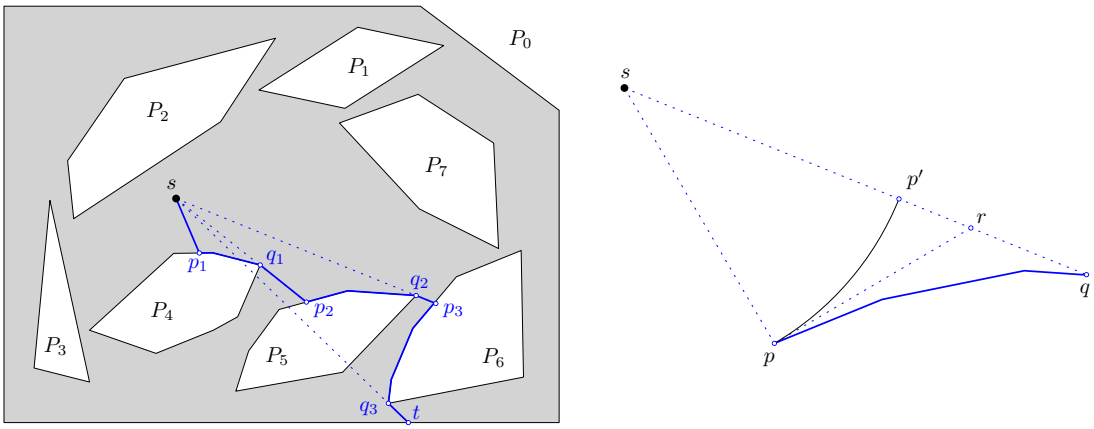


Figure 1: Left: A polygon $P \in \mathcal{C}(7)$ with 7 convex holes, a point $s \in P$, and a path $\text{greedy}_P(s)$ from s to a point t on the outer boundary of P. Right: A boundary arc $\widehat{pq}$, where $|\widehat{pq}| \leq |pr| + |rq|$.

We construct a path $\text{greedy}_P(s)$ from s to some point t in the outer boundary of P by the following recursive algorithm; refer to Fig. 1 (left): Start from s along an arbitrary ray emanating from s until reaching the boundary of P at some point p. If $p \in \partial P_0$, then let $t = q$ and the path $\text{greedy}_P(s)$ terminates at t. Otherwise $p \in \partial P_i$, $1 \le i \le h$, and γ follows ∂P_i to the point ℓ_i or r_i such that the distance from s monotonically increases; and then continues along the ray $\overrightarrow{s\ell_i}$ or $\overrightarrow{sr_i}$ until hits the boundary of P again.

Lemma 3 *For every $P \in \mathcal{C}(h)$ and every point $s \in P$, we have $|\text{greedy}_P(s)| \le O(h^{1/2}) \cdot \text{diam}_2(P)$.*

Proof: Let P be a polygonal domain with a convex outer polygon P_0 and h convex holes. We may assume w.l.o.g. that $\text{diam}_2(P) = 1$. For a point $s \in P$, consider the path $\text{greedy}_P(s)$. By construction, the distance from s monotonically increases along $\text{greedy}_P(s)$, and so the path has no self-intersections. It it composed of *radial segments* that lie along rays emanating from s, and *boundary arcs* that lie on the boundaries of holes. By monotonicity, the total length of all radial segments is at most $\text{diam}_2(P)$. Since every boundary arc ends at a point of tangency ℓ_i or r_i, for some $i \in \{1, \ldots, h\}$, then $\text{greedy}_P(s)$ contains at most two boundary arcs along each hole, thus number of boundary arcs is at most $2h$. Let $\mathcal{A}$ denote the set of all boundary arcs along $\text{greedy}_P(s)$; then $|\mathcal{A}| \le 2h$.

Along each boundary arc $\widehat{pq} \in \mathcal{A}$, from p to q, the distance from s increases by $\Delta_{pq} = |sq| - |sp|$. By monotonicity, we have

$$\sum_{\widehat{pq} \in \mathcal{A}} \Delta_{pq} \le \text{diam}_2(P).$$

We now give an upper bound for the length of $\widehat{pq}$. Let p' be a point in sq such that $|sp| = |sp'|$, and let r be the intersection of sq with a line orthogonal to sp passing through p; see Fig. 1 (right). Note that $|sp| < |sr|$. Since the distance from s monotonically increases along the arc $\widehat{pq}$, then q is in the closed halfplane bounded by pr that does not contain s. Combined with $|sp| < |sr|$, this implies that r lies between p' and q on the line sq, consequently $|p'r| < |p'q| = \Delta_{pq}$ and $|rq| < |p'q| = \Delta_{pq}$. By the triangle inequality and the Pythagorean theorem, these estimates give an upper bound

$$\begin{aligned}
|\widehat{pq}| &\le |pr| + |rq| = \sqrt{|sr|^2 - |sp|^2} + |rq| \le \sqrt{(|sp'| + |p'r|)^2 - |sp|^2} + |rq| \\
&\le \sqrt{(|sp| + \Delta_{pq})^2 - |sp|^2} + \Delta_{pq} \le O\left(\sqrt{|sp|\Delta_{pq}} + \Delta_{pq}\right) \\
&\le O\left(\sqrt{\text{diam}_2(P) \cdot \Delta_{pq}} + \Delta_{pq}\right).
\end{aligned}$$

Summation over all boundary arcs, using Jensen's inequality, yields

$$\begin{aligned}
\sum_{\widehat{pq} \in \mathcal{A}} |\widehat{pq}| &\le \sum_{\widehat{pq} \in \mathcal{A}} O\left(\sqrt{\text{diam}_2(P) \cdot \Delta_{pq}} + \Delta_{pq}\right) \\
&\le \sqrt{\text{diam}_2(P)} \cdot O\left(\sum_{\widehat{pq} \in \mathcal{A}} \sqrt{\Delta_{pq}}\right) + O\left(\sum_{\widehat{pq} \in \mathcal{A}} \Delta_{pq}\right) \\
&\le \sqrt{\text{diam}_2(P)} \cdot O\left(|\mathcal{A}| \cdot \sqrt{\frac{1}{|\mathcal{A}|} \sum_{\widehat{pq} \in \mathcal{A}} \Delta_{pq}}\right) + O(\text{diam}_2(P)) \\
&\le \sqrt{\text{diam}_2(P)} \cdot O\left(\sqrt{|\mathcal{A}| \cdot \text{diam}_2(P)}\right) + O(\text{diam}_2(P)) \\
&\le O\left(\sqrt{|\mathcal{A}|}\right) \cdot \text{diam}_2(P) \le O\left(\sqrt{h}\right) \cdot \text{diam}_2(P),
\end{aligned}$$

as claimed. □

Corollary 4 *For every $h \in \mathbb{N}$ and every polygon $P \in \mathcal{C}(h)$, we have $\text{diam}_g(P) \le O(h^{1/2}) \cdot \text{diam}_2(P)$.*

PROOF: Let $P \in \mathcal{C}(h)$ and $s_1, s_2 \in P$. By Lemma 3, there exist points $t_1, t_2 \in \partial P_0$ such that $\mathrm{geod}(s_1, t_1) \leq O(h^{1/2}) \cdot \mathrm{diam}_2(P)$ and $\mathrm{geod}(s_2, t_2) \leq O(h^{1/2}) \cdot \mathrm{diam}_2(P)$. There is a path between t_1 and t_2 along the perimeter of P_0, hence $\mathrm{geod}(t_1, t_2) \leq O(\mathrm{diam}_2(P))$. The concatenation of these three paths yields a path in P connecting s_1 and s_2, of length $\mathrm{geod}(s_1, s_2) \leq O(h^{1/2}) \cdot \mathrm{diam}_2(P)$, as required. □

Lower Bound. The lower bound in Theorem 1 is based on the following construction.

Lemma 5 *For every $h \in \mathbb{N}$, there exists a polygon $P \in \mathcal{C}(h)$ such that $g(P) \geq \Omega(h^{1/3})$.*

PROOF: We may assume w.l.o.g. that $h = k^3$ for some integer $k \geq 3$. We construct a polygon P with h holes, where the outer polygon P_0 is a regular k-gon of unit diameter, hence $\mathrm{diam}_2(P) = \mathrm{diam}_2(P_0) = 1$. Let $Q_0, Q_1, \ldots, Q_{k^2}$ be a sequence of $k^2 + 1$ regular k-gons with a common center such that $Q_0 = P_0$, and for every $i \in \{1, \ldots, k^2\}$, Q_i is inscribed in Q_{i-1} such that the vertices of Q_i are the midpoints of the edges of Q_{i-1}; see Fig. 2. Enumerate the k^3 edges of $Q_1, \ldots, Q_{k^2}$ as $e_1, \ldots, e_{k^3}$. For every $j = 1, \ldots, k^3$, we construct a hole as follows: Let P_j be an $(|e| - 2\varepsilon) \times \frac{\varepsilon}{2}$ rectangle with symmetry axis e that contains e with the exception of the ε-neighborhoods of its endpoints. Then $P_1, \ldots, P_{k^3}$ are pairwise disjoint. Finally, let $P = P_0 \setminus \bigcup_{j=1}^{k^3} P_j$.

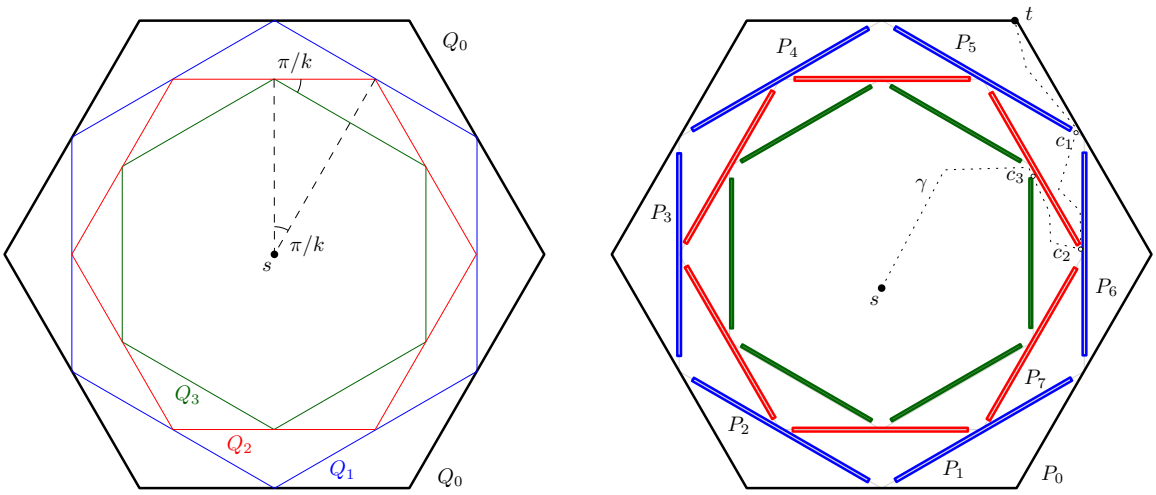


Figure 2: Left: hexagons $Q_0, \ldots, Q_3$ for $k = 6$. Right: The 18 holes corresponding to the edges of $Q_1, \ldots, Q_3$.

Assume, w.l.o.g., that e_i is an edge of Q_i for $i \in \{0, 1, \ldots, k^2\}$. As $P_0 = Q_0$ is a regular k-gon of unit diameter, then $|e_0| \geq \Omega(1/k)$. Let us compare the edge lengths in two consecutive k-gons. Since Q_{i+1} is inscribed in Q_i, we have

$$|e_{i+1}| = |e_i| \cos \frac{\pi}{k} \geq |e_i| \left(1 - \frac{\pi^2}{2k^2}\right)$$

using the Taylor estimate $\cos x \geq 1 - x^2/2$. Consequently, for every $i \in \{0, 1 \ldots, k^2\}$,

$$|e_i| \geq |e_0| \cdot \left(1 - \frac{\pi^2}{2k^2}\right)^{k^2} \geq |e_0| \cdot \Omega(1) \geq \Omega\left(\frac{1}{k}\right).$$

It remains to show that $\mathrm{diam}_g(P) \geq \Omega(k)$. Let s be the center of P_0 and t and arbitrary vertex of P_0. Consider an st-path γ in P, and for any two points a, b along γ, let $\gamma(a, b)$ denote the subpath of γ between a and b. Let c_i be the first point where γ crosses the boundary of Q_i for $i \in \{1, \ldots k^2\}$. By construction, c_i must be in an ε-neighborhood of a vertex of Q_i. Since the vertices of Q_{i+1} are at the midpoints of the edges of Q_i, then $|\gamma(c_i, c_{i+1})| \geq \frac{1}{2}|e_i| - 2\varepsilon \geq \Omega(|e_i|) \geq \Omega(1/k)$. Summation over $i = 0, \ldots, k^2 - 1$ yields $|\gamma| \geq \sum_{i=0}^{k^2-1} |\gamma(c_i, c_{i+1})| \geq k^2 \cdot \Omega(1/k) \geq \Omega(k) = \Omega(h^{1/3})$, as required. □

3 Convex Polygons with Fat Convex Holes

In this section, we show that in a polygonal domain P with fat convex holes, the distortion $\text{geod}(s,t)/|st|$ is bounded by a constant for all $s,t \in P$, and prove Proposition 2.

Let C be a convex body C in the plane, and let $P = \mathbb{R}^2 \setminus C$ be its complement. For any two points $s,t \in \partial C$, we compare the Euclidean distance $|s,t|$ with the geodesic distance $\text{geod}(s,t)$, which is the shortest st-path along the boundary of C. The *geometric dilation* of C is $\delta(C) = \sup_{s,t\in\partial C} \frac{\text{geod}(s,t)}{|st|}$.

Lemma 6 *Let C be a λ-fat convex body. Then $\delta(C) \leq \min\{\pi\lambda^{-1}, 2(\lambda^{-1}+1)\} = O(\lambda^{-1})$.*

Proof: It is known [6, Lemma 11] that $\delta(C) = \frac{|\partial C|}{2h}$, where $h = h(C)$ is the *minimum halving distance* of C (i.e., the minimum distance between two points on C that divide the length of C in two equal parts). It is also known [5, Thm. 8] that $h \geq \text{width}(C)/2$. Putting these together one deduces that $\delta(C) \leq \frac{|C|}{\text{width}(C)}$. The isoperimetric inequality $|\partial C| \leq \text{diam}_2(C)\pi$ and the obvious inequality $|\partial C| \leq 2\text{diam}_2(C)+2\text{width}(C)$ lead to the following dilation bounds $\delta(C) \leq \pi \frac{\text{diam}_2(C)}{\text{width}(C)}$ and $\delta(C) \leq 2\left(\frac{\text{diam}_2(C)}{\text{width}(C)}+1\right)$; see also [5, 15]. Since C is λ-fat, direct substitution yields the two bounds given in the lemma. Note that the latter bound is better for small λ. □

Corollary 7 *Let $P = P_0 \setminus \left(\bigcup_{i=1}^h P_i\right)$ be a polygonal domain, where $P_0, P_1, \ldots, P_h$ are λ-fat convex polygons. Then for any $s,t \in P$, we have* $\text{geod}(s,t) \leq O(\lambda^{-1}|st|)$.

Proof: If the line segment st is contained in P, then $\text{geod}(s,t) = |st|$, and the proof is complete. Otherwise, segment st is the concatenation of line segments contained in P and line segments $p_iq_i \subset P_i$ with $p_i, q_i \in \partial P_i$, for some indices $i \in \{1,\ldots,h\}$. By replacing each segment p_iq_i with the shortest path on the boundary of the hole P_i, we obtain an st-path γ in P. Since each hole is λ-fat, we replaced each line segment p_iq_i with a path of length $O(|p_iq_i|/\lambda)$ by Lemma 6. Overall, we have $|\gamma| \leq O(|st|/\lambda)$, as required. □

Corollary 8 *If $P = P_0 \setminus \left(\bigcup_{i=1}^h P_i\right)$ be a polygonal domain, where $P_0, P_1, \ldots, P_h$ are λ-fat convex polygons for some $0 < \lambda \leq 1$, then $\text{diam}_g(P) \leq O(\lambda^{-1}\text{diam}_2(P))$, hence $\varrho(P) \leq O(\lambda^{-1})$.*

4 Conclusion

We have shown that in a convex polygonal domain P with h convex holes, the distortion of the polygon, $\varrho(P) = \frac{\text{diam}_g(P)}{\text{diam}_2(P)}$, is always $O(h^{1/2})$ and sometimes $\Omega(h^{1/3})$. The following version of the question studied here may be more attractive to the escape community. Given n pairwise disjoint convex obstacles in a convex polygon of diameter $O(1)$ (e.g., the unit square), what is the maximum length of a (shortest) escape route from any given point in the polyon to the polygon's boundary? According to our results, it is always $O(n^{1/2})$ and sometimes $\Omega(n^{1/3})$.

Closing the gap between the upper and lower bounds is work in progress. Generalizations to d-dimensional Euclidean spaces for $d \geq 3$ are left for future research. It would also be interesting to improve the running time of algorithms for computing the geodesic diameter or radius of a polygon with h holes when all holes as well as the outer polygon are convex.

References

[1] Hee-Kap Ahn, Luis Barba, Prosenjit Bose, Jean-Lou De Carufel, Matias Korman, and Eunjin Oh, A linear-time algorithm for the geodesic center of a simple polygon. *Discrete & Computational Geometry*, 56:836–859, 2016. doi:10.1007/s00454-016-9796-0.

[2] Sang Won Bae, Matias Korman, and Yoshio Okamoto, The geodesic diameter of polygonal domains. *Discrete & Computational Geometry*, 50(2):306–329, 2013. doi:10.1007/s00454-013-9527-8.

[3] Sang Won Bae, Matias Korman, and Yoshio Okamoto. Computing the geodesic centers of a polygonal domain. *Comput. Geom.*, 77:3–9, 2019. doi:10.1016/j.comgeo.2015.10.009.

[4] Yi-Jen Chiang and Joseph S. B. Mitchell, Two-point Euclidean shortest path queries in the plane. In *Proc. 10th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 215–224, 1999. URL: https://dl.acm.org/doi/10.5555/314500.314560.

[5] Adrian Dumitrescu, Annette Ebbers-Baumann, Ansgar Grüne, Rolf Klein, and Günter Rote, On the geometric dilation of closed curves, graphs, and point sets. *Comput. Geom.*, 36(1):16–38, 2007. doi:10.1016/j.comgeo.2005.07.004.

[6] Annette Ebbers-Baumann, Ansgar Grüne, and Rolf Klein, Geometric dilation of closed planar curves: New lower bounds. *Comput. Geom.*, 37(3):188–208, 2007. doi:10.1016/j.comgeo.2004.12.009.

[7] Leonidas J. Guibas and John Hershberger, Optimal shortest path queries in a simple polygon. *J. Comput. Syst. Sci.*, 39:126–152, 1989. doi:10.1016/0022-0000(89)90041-X.

[8] Sariel Har-Peled, Shortest path in a polygon using sublinear space. *J. Comput. Geom.*, 7:19–45, 2015. doi:10.20382/jocg.v7i2a3.

[9] John Hershberger and Subhash Suri, Matrix searching with the shortest-path metric. *SIAM J. Computing*, 26(6):1612–1634, 1997. doi:10.1137/S0097539793253577.

[10] John Hershberger and Subhash Suri, An optimal algorithm for Euclidean shortest paths in the plane. *SIAM J. Computing*, 28(6):2215–2256, 1999. doi:10.1137/S0097539795289604.

[11] Sunjiv Kapoor, Shachindra N. Maheshwari, and Joeseph S. B. Mitchell, An efficient algorithm for Euclidean shortest paths among polygonal obstacles in the plane. *Discrete & Computational Geometry*, 18:377–383, 1997. doi:10.1007/PL00009323.

[12] Der-Tsai Lee and Franco P. Preparata, Euclidean shortest paths in the presence of rectilinear barriers. *Networks*, 14:393–410, 1984. doi:10.1002/net.3230140304.

[13] Joseph S. B. Mitchell, Shortest paths among obstacles in the plane. *Int. J. Comput. Geom. Appl.*, 6(3):309–332, 1996. doi:10.1142/S0218195996000216.

[14] Joseph S.B. Mitchell, Shortest paths and networks. In *Handbook of Discrete and Computational Geometry*, chapter 31. CRC Press, Boca Raton, FL, 3 edition, 2017. doi:10.1201/9781315119601.

[15] Paul R. Scott and Poh Wah Awyong, Inequalities for convex sets. *Journal of Inequalities in Pure and Applied Mathematics*, 1:article 6, 2000.

[16] Haitao Wang, On the geodesic centers of polygonal domains. *J. Comput. Geom.*, 9(1):131–190, 2018. doi:10.20382/jocg.v9i1a5.

[17] Haitao Wang, A new algorithm for Euclidean shortest paths in the plane. In *Proc. 53rd ACM Symposium on Theory of Computing (STOC)*, pages 975–988, 2021. doi:10.1145/3406325.3451037.

[18] Haitao Wang, Shortest paths among obstacles in the plane revisited. In *Proc. 32nd ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 810–821, 2021. doi:10.1137/1.9781611976465.51.

Helly-type theorems for hypergraphs

Csaba Biró

Department of Mathematics
University of Louisville,
Louisville, KY 40292, USA
csaba.biro@louisville.edu

Jenő Lehel

Department of Mathematics
University of Louisville,
Louisville, KY 40292, USA
jeno.lehel@louisville.edu

Géza Tóth[1]

Rényi Institute of Mathematics and
Budapest University of Technology and
Economics, SZIT, Hungary
geza@renyi.hu

Abstract: Let H be a complete r-uniform hypergraph such that two vertices are marked in each edge as its 'boundary' vertices. A linear ordering of the vertex set of H is called an *agreeing linear order*, provided all vertices of each edge of H lie between its two boundary vertices. We prove the following Helly-type theorem: if there is an agreeing linear order on the vertex set of every subhypergraph of H with at most $2r-2$ vertices, then there is an agreeing linear order on the vertex set of H. We also show that the constant $2r-2$ cannot be reduced in the theorem. The case $r=3$ of the theorem has particular interest in the axiomatic theory of betweenness. Similar results are obtained for further r-uniform hypergraphs ($r \geq 3$), where one or two vertices are marked in each edge, and the linear orders need to satisfy various rules of agreement. In one of the cases we prove that no such Helly-type statement holds.

Keywords: Helly-type theorem, r-uniform clique, agreeing linear order

An ordered hypergraph is a hypergraph together with a linear ordering of its vertex set. These combinatorial structures emerge in various contexts, from the study of matrices with forbidden submatrices [1], to the modeling combinatorial geometry problems [5]. Here we are dealing with combinatorial problems on ordered hypergraphs originated in the study of betweenness of convex bodies in the plane [3].

Let H be a complete uniform hypergraph with no repeated edges, that is a *clique*. If every edge $e \in E(H)$ has a set of two 'marked' vertices, denoted ∂e, then H is called here a 2-*extreme marked clique*. A linear order L on $V(H)$ is called an *agreeing linear order* of H, if for every $e \in E(H)$ the set of the L-minimal and the L-maximal elements in e is equal to ∂e. Similarly, for a set of vertices $U \subseteq V(H)$, a linear order L on U is called an *agreeing linear order* of U, if for every $e \in E(H)$, $e \subseteq U$, the L-minimal and the L-maximal elements in e is equal to ∂e.

We might say that a 2-extreme marked clique is an ordered hypergraph with a vertex set ordered according to an agreeing linear order. We prove here a Helly-type theorem on the existence of an agreeing linear order for 2-extreme marked cliques as follows.

Theorem 1 *Let H be a 2-extreme marked r-uniform clique on at least $2r-2$ vertices ($r \geq 3$). If every subset of $2r-2$ vertices of $V(H)$ has an agreeing linear order, then H has an agreeing linear order. Furthermore, there exists a clique H such that every set of $2r-3$ vertices of H has an agreeing linear order, but H does not.*

[1]Supported by National Research, Development and Innovation Office, NKFIH, K-131529 and ERC Advanced Grant "GeoScape," No. 882971.

In Section 1, we prove Theorem 1 by using two different approaches. Section 1.1 proves the case $r = 3$ separately (Theorem 4); we could not find a convenient extension of this 'direct' proof for $r \geq 4$. Section 1.2 contains a proof for $r \geq 4$ (Theorem 10 through Lemmas 5 and 7); this approach does not work for $r = 3$ without making the proof of Lemma 7 much less transparent.

Actually, a 3-uniform 2-extreme marked clique H can be considered as 'one-marked' by marking the 'middle' vertex $\widehat{e} = e \setminus \partial e$ for every $e \in E(H)$; and an agreeing linear order requires that $\widehat{e}$ be positioned between the two vertices of ∂e, for every $e \in E(H)$. In Sections 2, 3, and 4 we extend the concept of ordered hypergraphs, where one or two vertices are marked in each edge, and the marked vertices agree with particular rules in the linear ordering of the vertex set.

Given an r-uniform clique with edges containing one or two marked vertices, our main interest consists in finding Helly-type theorems guaranteeing the existence of an agreeing linear order for the hypergraph. We supply a table of content of the paper that may help the reader familiarize the non-standard notions.

clique	**marked**	**agreeing lin. ord.**	**Helly-number**	
2-extreme marked	∂e	∂e **has the min. and max. of** e	$2r-2$	**Section 1**
min-marked	$A(e)$	$A(e)$ **is the min. in** e	$r+1$	**Section 2**
1-extreme marked	$\widehat{e}$	$\widehat{e}$ **is the min. or max. in** e	—	**Section 3**
min&max-marked	$\{A(e), B(e)\}$	$A(e)$ **is min.** $B(e)$ **is max. in** e	$2r-2$	**Section 4**

In Section 2, we consider r-uniform min-marked cliques, with one vertex $A(e)$ marked in each edge e, and in an agreeing linear order $A(e)$ is minimal among the vertices of e. We obtain a Helly-type theorem whose proof reveals a characterization of min-marked ordered hypergraphs H that have an agreeing linear order, in general, in terms of a forbidden 2×2 submatrix in the incidence matrix of H (Theorem 11).

Theorem 2 *Let H be a min-marked r-uniform clique on at least $r+1$ vertices ($r \geq 3$). Then the vertices of H have an agreeing linear order if and only if each subhypergraph of H with $r+1$ vertices has an agreeing linear order.*

In Section 3 the agreeing linear order of a 1-extreme marked hypergraph is investigated that requires the marked vertex $\widehat{e}$ be either the minimal or the maximal among the vertices in each edge e. A straightforward characterization (Proposition 12) leads to the unexpected fact that there is no Helly-type theorem for the existence of an agreeing linear order for 1-extreme marked cliques (Proposition 13).

Another Helly-type result concludes the paper by characterizing those min&max-marked cliques which admit an agreeing linear order where both the minimum and the maximum vertices are prescribed in each edge.

Theorem 3 *Let H be a min&max-marked r-uniform clique with at least $2r-2$ vertices ($r \geq 3$). Then H has an agreeing linear order if and only if each subhypergraph of H with $2r-2$ vertices has an agreeing linear order.*

Theorem 3 can be deduced as a corollary of Theorem 1; an independent graph theory proof is also given in Section 4.

1 2-extreme marked cliques

Let H be an r-uniform clique. If every edge of H has two marked vertices, then H is called here a 2-*extreme marked clique*. The name '2-extreme marked' indicates that the marked vertices of an edge should become the maximum and the minimum among the vertices of each edge in an agreeing linear order. The set of the two marked vertices of an edge e is denoted by ∂e.

Let $U \subseteq V(H)$ and let L be a linear ordering on U. We say that an edge $\{v_1, \ldots, v_r\} \subset U$ *agrees* with L, provided $v_1 <_L v_2 <_L \cdots <_L v_r$ and $\partial e = \{v_1, v_r\}$; furthermore, L is called an *agreeing linear order* on U, if every edge entirely in U agrees with L.

In the next sections we prove Theorem 1, a Helly-type theorem on the existence of an agreeing linear order for 2-extreme marked cliques. The special case $r = 3$ is proved separately in Section 1.1. It is worth noting that we could not find a convenient extension of the 'direct' proof of this special case for $r \geq 4$. The proof of the general case in Section 1.2 uses a different approach.

1.1 The case $r = 3$

Here we prove the special case $r = 3$ of Theorem 1.

Theorem 4 *Let H be a 2-extreme marked 3-uniform clique. If every set of 4 vertices of H has an agreeing linear order, then H has an agreeing linear order.*

PROOF: Let $x \in V(H)$. We define a binary relation $\sim = \sim_x$ on $V(H) \setminus \{x\}$ with respect to x: let $u \sim v$ if $u = v$ or $u \neq v$ and $\partial\{x, u, v\} \neq \{u, v\}$. We show that the relation $\sim$ is an equivalence relation.

Only transitivity is nontrivial. Let $u \sim v$ and $v \sim w$. Suppose for a contradiction that $\partial\{x, u, w\} = \{u, w\}$. The set $\{x, u, v, w\}$ has an agreeing linear order L. In L, we have x between u and w; due to symmetry we may assume $u <_L x <_L w$. So where is v? Since $u \sim v$, we must have $v <_L x$, but since $v \sim w$, we must have $v >_L x$, a contradiction.

We claim that there are at most two equivalence classes of the relation $\sim$. We will prove this by showing that $u \not\sim v$ and $v \not\sim w$ imply $u \sim w$. Suppose this is not true, and chose vertices with $u \not\sim v$, $v \not\sim w$, and $u \not\sim w$. The set $\{x, u, v, w\}$ has an agreeing linear order L. Similarly as above, we may assume $u <_L x <_L w$. This time, $u \not\sim v$ implies $v >_L x$, and $v \not\sim w$ implies $v <_L x$ in L, a contradiction. Note also that if x is not in the boundary of some edge, say $x \notin \partial\{x, u, v\}$, then there are exactly two equivalence classes, since $\partial\{x, u, v\} = \{u, v\}$ implies $u \not\sim v$.

The proof of the theorem proceeds by induction on $|V(H)|$. If $|V(H)| = 2$, the statement is trivial. Let $|V(H)| \geq 3$, and chose a vertex x that is not in the boundary of some edge. Let the two equivalence classes with respect to x be A_1 and A_2. By the hypothesis, $A_i \cup \{x\}$ has an agreeing linear order L_i for $i = 1, 2$.

Observe that x is the greatest or the least element of L_1 and L_2. Indeed, if $u, v \in A_i$ was such that $u < x < v$ in L_i, then $\partial\{x, u, v\} = \{u, v\}$, so $u \not\sim v$ contradicting $u, v \in A_i$. After possibly taking duals, we may assume that x is the greatest element of L_1 and the least element of L_2. Concatenate L_1 and L_2 by adding every relation $A_1 < A_2$ to form the linear order L on $V(H)$.

We conclude the proof by showing that L is an agreeing linear order. Let $e \in E(H)$. If $e \subseteq A_i \cup \{x\}$ for some i, then e agrees with L. If $e = \{u, x, v\}$ with $u \in A_1$ and $v \in A_2$, then $u \not\sim v$ exactly means $\partial\{u, x, v\} = \{u, v\}$, so e agrees with L.

The remaining case (up to symmetry) is $e = \{u, v, w\}$, $u, v \in A_1$, $u < v$ in L_1, and $w \in A_2$. The set $\{u, v, w, x\}$ has an agreeing linear order L'. Since $u \not\sim w$, we may assume (up to duality) that $u < x < w$ in L'. Since $v \not\sim w$, we have $v < x$ in L'. Note that $u < v < x$ in L_1 implies $\partial\{u, v, x\} = \{u, x\}$, so in L', we must have $u < v < x < w$. Since L' is agreeing, this shows $\partial\{u, v, w\} = \{u, w\}$, and thus e agrees with L. □

1.2 The case $r \geq 4$

Here we restate Theorem 1 and prove it for $r \geq 4$. The proof uses two lemmas.

Lemma 5 *Let $r \geq 4$, and assume that L is an agreeing linear order for the 2-extreme marked r-uniform clique H with n vertices. Then L is unique (up to duality) if and only if $n \geq 2r - 3$.*

PROOF: Let $(u_1, \ldots, u_n)$ be an agreeing linear order of the vertices of H. If $n \leq 2r - 4$ and $r \geq 4$, then $u_{\lfloor n/2 \rfloor}$ and $u_{\lfloor n/2 \rfloor + 1}$ are not boundary vertices in any edge. Therefore, they can be swapped to obtain another agreeing linear order. Because the agreeing linear order is not unique, $n \geq 2r - 3$ follows.

Let $n \geq 2r - 3$. Let $L_1 = (u_1, \ldots, u_n)$ and L_2 be agreeing linear orders of H.

Claim 6 *There are no vertices u_i, u_j with $i < j$ such that*

- *if $u_1 <_{L_2} u_n$ then $u_i >_{L_2} u_j$;*
- *if $u_1 >_{L_2} u_n$ then $u_i <_{L_2} u_j$.*

PROOF: Suppose such vertices exist. We may assume that $i \neq 1$ and $j \neq n$, otherwise the statement is trivial, since every $u_i, u_j \notin \{u_1, u_n\}$ is between u_1 and u_n in L_2. Observe that because $(j-1)+(n-i) \geq n \geq 2r-3$, we have

$$j \geq r \qquad \text{or} \qquad i \leq n-r+1. \tag{1}$$

Assume first $u_1 <_{L_2} u_n$. Let $A = \{v_1, \ldots, v_{r-4}\}$ be a (possibly empty) subset of $V(H)$ such that $u_1, u_i, u_j, u_n \notin A$, and let $e = A \cup \{u_1, u_i, u_j, u_n\}$. Since $\partial e = \{u_1, u_n\}$, we have that u_i and u_j are both between u_1 and u_n in L_2, so specifically,

$$u_1 <_{L_2} u_j <_{L_2} u_i <_{L_2} u_n. \tag{2}$$

Recalling (1), suppose first that $j \geq r$. This means that there exists an edge $f \subseteq \{u_1, \ldots, u_j\}$ such that $\{u_1, u_i, u_j\} \subseteq f$. Since $\partial f = \{u_1, u_j\}$, it follows that u_i is between u_1 and u_j in L_2, contradicting (2). Now suppose that $i \leq n-r+1$. Then there exists $g \subseteq \{u_i, \ldots, u_n\}$ such that $\{u_i, u_j, u_n\} \subseteq g$. Since $\partial g = \{u_i, u_n\}$, it follows that u_j is between u_i and u_n in L_2, contradicting (2).

Following a similar argument for $u_1 >_{L_2} u_n$, equation (2) turns into

$$u_n <_{L_2} u_i <_{L_2} u_j <_{L_2} u_1,$$

and a similar argument to the one above leads to a contradiction. □

It is now easy to see how this technical claim implies the Lemma. If $u_1 <_{L_2} u_n$, then every pair of vertices is ordered the same in L_2 as in L_1, so $L_1 = L_2$. If $u_1 >_{L_2} u_n$, then every pair of vertices is ordered the opposite in L_2 as in L_1, so $L_2 = L_1^d$. □

A vertex $\ell \in V(H)$ is an *extremal* vertex of a 2-extreme marked clique H, if every $e \in E(H)$ containing ℓ satisfies $\ell \in \partial e$. Note that H has at most two extremal vertices.

Lemma 7 *For $r \geq 4$, let H be a 2-extreme marked r-uniform clique with $|V(H)| \geq 2r-2$. If every set of $2r-2$ vertices of H has an agreeing linear order, then H has exactly two extremal vertices.*

PROOF: We proceed by induction on the number n of vertices. If $n = 2r-2$, then H has an agreeing linear order L. The least element ℓ_1 and the greatest element ℓ_2 of L are clearly extremal vertices of H.

Now let $n > 2r-2$, and suppose that the lemma is true for $n-1$. Let $x \in V(H)$ be a non-extremal vertex of H, and let $H' = H - x$. We apply the hypothesis to find extremal vertices ℓ_1 and ℓ_2 in H'. Next consider the edge set

$$F = \{\{x, \ell_1, \ell_2, v_1, \ldots, v_{r-3}\} : v_1, \ldots, v_{r-3} \in V(H) \setminus \{x, \ell_1, \ell_2\}\}.$$

Claim 8 *For some $f \in F$ we have $x \notin \partial f$.*

PROOF: Since x is not extremal in H, we have $x \notin \partial e$ for some $e \in E$. Let $S \subset V(H)$ be a set containing $e \cup \{\ell_1, \ell_2\}$ such that $r+1 \leq |S| \leq r+2 \leq 2r-2$. Then S has an agreeing linear order L. Because $x \notin \partial e$, vertex x is not the minimal or maximal element in S.

Suppose that some $a \notin \{x, \ell_1, \ell_2\}$ is the minimal or maximal element of S. Take a subset $g \subseteq S$, $|g| = r$, $x \notin g$. Since L is an agreeing linear order on g, one of the extremal vertices of g is a, which is a contradiction, since $a \in g \subset V(H) \setminus \{x\}$ is between ℓ_1 and ℓ_2, the extremal vertices of g. Therefore, ℓ_1 and ℓ_2 are the minimal and maximal elements of S, so we have $x \notin \partial f$ for some $f \in F$. □

Claim 9 *For every $f \in F$ we have $x \notin \partial f$.*

PROOF: By Claim 8, $x \in \partial f$ for some $f \in F$. Assume for a contradiction that $x \notin \partial f'$ for some $f' \in F$. Let these edges be

$$f = \{x, \ell_1, \ell_2, v_1, \ldots, v_{r-3}\}, \qquad f' = \{x, \ell_1, \ell_2, v'_1, \ldots, v'_{r-3}\}.$$

Notice that this step assumes $r \geq 4$. Since $|f \cup f'| \leq 2r-3$, we have an agreeing linear order L on $f \cup f'$.

Let W be any $r-2$-element subset of $V = \{v_1, \ldots, v_{r-3}, v'_1, \ldots, v'_{r-3}\}$. Then, by the hypothesis, $\partial(W \cup \{\ell_1, \ell_2\}) = \{\ell_1, \ell_2\}$. This means that in L, every element of V is between ℓ_1, and ℓ_2. On the other hand, due to $x \in \partial f'$, we have that x is not between ℓ_1 and ℓ_2. So up to symmetry and the possible exchange of ℓ_1 and ℓ_2, we obtain

$$\ell_1 <_L V <_L \ell_2 <_L x$$

Then $f \subset V \cup \{x, \ell_1, \ell_2\}$ implies $x \in \partial f$, a contradiction. □

We show that both ℓ_1 and ℓ_2 are extremal in $V(H)$. It is enough to show the statement for ℓ_1; the proof for ℓ_2 is similar. If ℓ_1 is not extremal, there exists an edge $f = \{\ell_1, x, v_1, \ldots, v_{r-2}\}$ for which $\ell_1 \notin \partial f$. By induction, the set $\{x, \ell_1, \ell_2, v_1, \ldots, v_{r-2}\}$ has an agreeing linear order L.

Let $V = \{v_1, \ldots, v_{r-2}\}$, and let W be any $r-3$-element subset of V. Since $\partial(V \cup \{\ell_1, \ell_2\}) = \{\ell_1, \ell_2\}$, we have that every element of V is between ℓ_1 and ℓ_2 in L. From the fact that $\ell_1 \notin \partial f$, some v_i and x must surround ℓ_1 in L. So up to symmetry,

$$x <_L \ell_1 <_L V <_L \ell_2$$

Since $e = W \cup \{x, \ell_1, \ell_2\} \in F$, this contradicts $x \notin \partial e$ proved in Claim 9. □

Theorem 10 *For $r \geq 4$, let H be a 2-extreme marked r-uniform clique with at least $2r-2$ vertices. Then H has an agreeing linear order if and only if every subhypergraph of H on $2r-2$ vertices has an agreeing linear order. Furthermore, the constant $2r-2$ cannot be replaced by a smaller value.*

PROOF: Necessity in the first claim is obvious, we prove the sufficiency by induction on the order of $H = (V, E)$. The claim is true for $|V| = 2r-2$, by the condition. Let $|V| = n+1$, $n \geq 2r-2$, and assume that the claim is true for n vertices. We may assume, by Lemma 7, that v_0, v_n are the extremal vertices of H. By induction, there is an agreeing linear order L for $H - v_0$. Up to duality, we may assume that $L = (v_1, \ldots, v_n)$, and this order is unique by Lemma 5. We claim that $(v_0, v_1, \ldots, v_n)$ is an agreeing linear order for H.

Let $e = \{v_0, v_{i_1}, v_{i_2}, \ldots, v_{i_{r-1}}\}$, $1 \leq i_1 < \ldots < i_{r-1} \leq n$, be an arbitrary edge of H. Consider a $(2r-2)$-element set U including e and containing v_n. By induction and due to Lemma 5, U has an agreeing linear order $L' = (v_0, \ldots, v_n)$ which is unique as well. Notice that $|U \cap (V \setminus \{v_0\})| = 2r-3 > r$; the union of L and L' yields the required agreeing linear order for H, and $\partial e = \{v_0, v_{i_{r-1}}\}$ follows.

To see the second claim of the theorem we present a hypergraph H on $n = 2r-2$ vertices, for every $r \geq 4$, such that the vertex set of each subhypergraph of H with $2r-3$ has an agreeing linear order but $V(H)$ does not. Let $V(H) = \{v_1, v_2, \ldots, v_{2r-2}\}$; for $e_0 = \{v_1, \ldots, v_{r-1}, v_r\}$ define $\partial e_0 = \{v_1, v_{r-1}\}$, and for every $e = \{v_{i_1}, v_{i_2}, \ldots, v_{i_r}\}$, $1 \leq i_1 < i_2 < \ldots i_r \leq 2r-2$, different from e_0 define $\partial e = \{v_{i_1}, v_{i_r}\}$. Let $e_1 = \{v_{r-1}, v_r, \ldots, v_{2r-2}\}$.

Observe that the extremal vertices of H are v_1 and v_{2r-2}. In any agreeing linear order $L = (v_1, \ldots, v_{2r-2})$ the first element of e_1 precedes the last element of e_0, that is $v_{r-1} <_L v_{r-1}$, a contradiction. Therefore an agreeing linear order can not exist.

Observe that $L_1 = (v_1, \ldots, v_{r-1}, v_r, \ldots, v_{2r-2})$ agrees with all edges of H but e_0, and $L_0 = (v_1, \ldots, v_r, v_{r-1}, \ldots, v_{2r-2})$ agrees with all edges of H but e_1. Because $e_0 \cup e_1 = V(H)$, no subhypergraph $H - v_i$ contains both e_0 and e_1; therefore, either $L_0 - v_i$ or $L_1 - v_i$ is an agreeing linear order of $V(H) \setminus \{v_i\}$, for every $1 \leq i \leq 2r-2$. □

2 Min-marked hypergraphs

Let H be an r-uniform hypergraph (not necessarily a clique), and let $A(e) \in e$ be the vertex marked in each $e \in E(H)$. This hypergraph will be called a *min-marked hypergraph* indicating that $A(e)$ should become the minimum among the vertices of each e in an agreeing linear order. A linear order L of H is called an *agreeing linear order*, provided $A(e) <_L v$, for every $e \in E(H)$ and $v \in e \setminus \{A(e)\}$.

Let $M(H)$ denote the incidence matrix of H, that is, rows correspond to the edges, columns correspond to the vertices, and for any $e \in E(H)$ and $\alpha \in V(H)$

$$m(e,\alpha) = \begin{cases} 0 & \text{if } \alpha \notin e \\ -1 & \text{if } \alpha = A(e) \\ 1 & \text{if } \alpha \in e \setminus \{A(e)\} \end{cases}$$

The matrix $F = \begin{array}{|c|c|}\hline \text{-1} & 1 \\ \hline 1 & \text{-1} \\ \hline \end{array}$ and its permutation will be called a *forbidden* 2×2. If L is an agreeing linear order of the min-marked hypergraph H, then the incidence matrix $M(H)$ contains no submatrix equivalent to a forbidden 2×2.

Theorem 2 is a direct consequence of the following characterication. Its proof is omitted in this extended abstract.

Theorem 11 *For a min-marked r-uniform clique H with at least $r+1$ vertices ($r \geq 3$) the following statements are equivalent*

(i) $M(H)$ contains no forbidden 2×2,

(ii) There is an agreeing linear order on every $(r+1)$-element subset of $V(H)$.

(iii) H has an agreeing linear order.

3 1-extreme marked hypergraphs

Let H be an r-uniform hypergraph (not necessarily a clique), and let $\widehat{e} \in e$ be a dedicated vertex in each edge $e \in E(H)$. This hypergraph will be called a 1-*extreme marked hypergraph*. The name '1-extreme marked' indicates that the marked vertex of an edge should become either minimum or maximum among the vertices of each edge in an agreeing linear order. A linear order L of a set $U \subseteq V(H)$ is an *agreeing linear order* on U provided $\widehat{e}$ is either L-minimal or L-maximal among the vertices of e for every edge $e \subseteq U$. An agreeing linear order on $V(H)$ is also called an agreeing linear order of H.

The edge/vertex incidence matrix $M(H)$ is defined for every $e \in E(H)$ and $\alpha \in V(H)$ by the entries

$$m(e,\alpha) = \begin{cases} 0 & \text{if } \alpha \notin e \\ -1 & \text{if } \alpha = \widehat{e} \\ 1 & \text{if } \alpha \in e \setminus \{\widehat{e}\} \end{cases}$$

3.1 Auxiliary graphs

Let H be a 1-extreme marked hypergraph, and let L be an agreeing linear order of H. A 2×2 submatrix of $M(H)$ equal to $\begin{array}{|c|c|}\hline \text{-1} & 1 \\ \hline 1 & \text{-1} \\ \hline \end{array}$ or its permutation is called an F-matrix; a 2×2 submatrix of $M(H)$ equal to $\begin{array}{|c|c|}\hline 1 & \text{-1} \\ \hline 1 & \text{-1} \\ \hline \end{array}$ or its permutation is called an S-matrix. The two vertices corresponding to the columns of an F-matrix are extremes of different types, one is the L-minimal, the other is the L-maximal element of the edges corresponding to the rows. Meanwhile, the column containing -1 of an S-matrix is either L-minimal or L-maximal element in both edges.

We associate a graph $SF(H)$ to H in two steps as follows. First let G be a graph with $V(G) = E(H)$ and for $e, f \in E(H)$ let ef be an edge in G labeled with S or F if and only if there exists an S-matrix or an F-matrix with rows e and f, respectively. Now $SF(H)$ is obtained from G by contracting all S-edges, then eliminating multiple F-edges and S-loops.

H	χ	α	β	γ	ξ
e	0	-1	1	1	0
f	0	1	-1	0	1
g	0	0	-1	1	1
h	1	0	1	-1	0
j	1	1	0	-1	0

Table 1:

As an example consider the hypergraph H with vertex set $V = \{\alpha, \beta, \gamma, \chi, \xi\}$ including the edges e, f, g, h and j marked as in Table 1. Then $SF(H)$ is a triangle on the compound vertices $\{f, g\}, \{h, j\}$ and $\{e\}$.

An agreeing linear order L of H defines a natural 2-coloring of the edges of H: color A or B will be assigned to $e \in E(H)$ if $\widehat{e}$ is L-minimal or L-maximal in e, respectively. In other words, $E(H) = E_A \cup E_B$, where E_A are edges of a min-marked subhypergraph of H and E_B are edges of a 'max-marked' subhypergraph of H, which is equivalent to a min-marked hypergraph.

To obtain a characterization similar to the one in Theorem 11 for min-marked hypergraphs, assume that $SF(H)$ is two-colorable (bipartite), we consider any proper two-coloring with A and B, and we associate to this A, B-coloring an auxiliary directed graph $AB(H)$ on vertex set $V(H)$ as follows. Each compound vertex represents a class of 'S-equivalent edges' of H; assign the color of a compound vertex γ of $SF(H)$ to all edges belonging to the class represented by γ. Thus we obtain a partition $E(H) = E_A \cup E_B$, where $E_A = \{e \in E(H) : \text{ if } e \text{ is colored with } A\}$ and $E_B = \{e \in E(H) : \text{ if } e \text{ is colored with } B\}$. For every $e \in E(H)$ and $\alpha, \beta \in e$, $\alpha \to \beta$ is an arc, if either $e \in E_A$ and $\widehat{e} = \alpha$, or $e \in E_B$ and $\widehat{e} = \beta$.

On the analogy of Theorem 11 we obtain a straightforward specification of 1-extreme marked hypergraphs in terms of auxiliary graphs.

Proposition 12 *A 1-extreme marked hypergraph H has an agreeing linear order if and only if $SF(H)$ contains no odd cycle and there is an $AB(H)$ graph associated with some proper two-coloring of $SF(H)$ that contains no directed cycle.*

Proof: Let L be an agreeing linear order of H. As described earlier, L defines a two-coloring, that is, a partition $E(H) = E_A \cup E_B$, where $E_A = \{e \in E(H) : \widehat{e} \text{ is } L\text{-minimal in } e\}$ and $E_B = \{e \in E(H) : \widehat{e} \text{ is } L\text{-maximal in } e\}$. Notice that the color is the same for all edges represented by any given compound vertex γ of $SF(H)$, therefore $SF(H)$ is a bipartite graph (and contains no odd cycle as required). Consider the $AB(H)$ graph associated with this two-coloring. Let $\alpha, \beta \in e$, for some $e \in E(H)$. If $\alpha \to \beta$ is an arc in $AB(H)$, then $\alpha <_L \beta$, because L agrees with e. Therefore, $AB(H)$ contains no directed cycle.

Next we prove that the requirements on the auxiliary graphs are sufficient for the existence of an agreeing linear order. Since $SF(H)$ is bipartite, its vertices have a proper two-coloring. Since $AB(H)$ associated with some A, B-coloring of $SF(H)$, it has no directed cycle, there is a linear ordering L on $V(H)$ such that $\alpha \to \beta$ implies $\alpha <_L \beta$. Observe that if $e \in E_A$ then for every $\beta \in e \setminus \{\widehat{e}\}$ we have $\widehat{e} \to \beta$, that is $\widehat{e}$ is L-minimal in e. Similarly, if $e \in E_B$ then for every $\alpha \in e \setminus \{\widehat{e}\}$ we have $\alpha \to \widehat{e}$, hence $\widehat{e}$ is L-maximal in e. □

3.2 1-extreme marked cliques

We will see here that the characterization of 1-extreme marked hypergraphs in Proposition 12 leads to

the somewhat unexpected outcome that no Helly-type theorem exists for 1-extreme marked cliques.

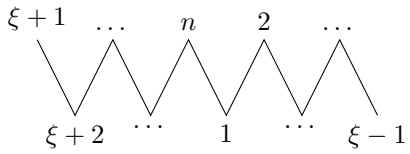
Proposition 13 *For every $r \geq 3$ and $n \geq r+1$ such that $n-r$ is even, there exists a 1-extreme marked r-uniform clique H on n vertices such that any subhypergraph of H on $n-1$ vertices has an agreeing linear order but H does not.*

Proof: We provide a construction for the case $r = 3$, which will be extended for every $r \geq 4$.

Case $r = 3$. Let $V(H) = \{1, 2, \ldots, n\}$, and let $e_i = \{i, i+1, i+2\}$ be edges with marked vertex $\widehat{e_i} = i+1$, for $i = 1, 2, \ldots, n$, (modulo n). For each 3-set $f = \{\alpha_1, \alpha_2, \alpha_3\} \subset V$, with $\alpha_1 < \alpha_2 < \alpha_3$, different from all e_i, $i = 1, 2, \ldots, n$, let $\widehat{f} = \alpha_1$.

Suppose an agreeing linear order L exists. Let C be the (odd) cycle on the vertex set $V(H)$ with edges $\{i, i+1\}$ (cyclically) for each i. The linear order L induces a 2-coloring on the edges as follows: $\{i, i+1\}$ is red, if $i >_L i+1$, and blue if $i <_L i+1$. The condition $\widehat{e_i} = i+1$ implies that this is a proper edge coloring of an odd cycle with 2 colors, a contradiction.

Now let $\xi \in V(H)$, and let $H' = H - \xi$. Organize the vertices of H' along a polygonal path, as shown below. (In this picture ξ is even. If ξ is odd, the only difference is the up/down position of the first and last point.)



Then list the lower elements on the picture in increasing order, and list the upper elements in decreasing order, i.e. $L = \{1, 3, \ldots, \xi-1, \xi+2, \xi+4, \ldots, n-1, n, n-2, \ldots, \xi+1, \xi-2, \ldots, 4, 2)$.

The linear order L clearly agrees with the edges $e_i \in E(H')$. For any other edge, if the marked vertex (i.e. lowest indexed vertex) is downstairs, it will be least in L; if it is upstairs, it will be greatest in L.

Case $r \geq 4$. We define a 1-extreme marked r-regular clique satisfying the requirements as follows. Notice that the condition $n - r \equiv 0 \pmod 2$ implies that $n_0 = n - r + 3$ is odd. Let $V_0 = \{1, 2 \ldots, n_0\}$, and let $V = V_0 \cup W$, where W is an $(r-3)$-set disjoint from V_0. Use the construction with vertex set $\{1, 2 \ldots, n_0\}$ as described above for the case $r = 3$ to obtain a 1-extreme marked 3-regular clique H_0 on vertex set V_0 with no agreeing linear order.

For every 3-set $f_0 \in E(H_0)$ define the r-set $f = f_0 \cup W$ with marked vertex $\widehat{f} = \widehat{f_0}$. For any other r-set $e \subset V$ we have $|e \cap V_0| \geq 4$; define $\widehat{e}$ to be the smallest value in $e \cap V_0$. Thus we obtain a 1-extreme marked r-uniform clique H.

Edges that contain 3 elements of V_0 prevent an agreeing linear order by a similar argument as above.

If $\xi \in W$ then $H - \xi$ becomes a min-marked hypergraph such that the ordering of the values $1, 2, \ldots, n_0$ followed by the elements of W in arbitrary order is an agreeing linear order of $H - \xi$.

For $1 \leq \xi \leq n_0$ an agreeing linear order for $H - \xi$ is obtained by using an agreeing linear order L_0 of $H_0 - \xi$, and by inserting the elements of W between $n_0 - 1$ and n_0 in any order. Equivalently, though less rigorously, W is inserted between the "downstairs" and the "upstairs" vertices of V_0. □

4 min&max-marked cliques

Let H be an r-uniform clique such that for every $e \in E(H)$ two distinct vertices are marked as a *min-vertex* and a *max-vertex*, denoted $A(e)$ and $B(e)$, respectively. This hypergraph will be called a *min&max-marked clique*.

A linear order L of the vertex set of a min&max-marked clique H is called an *agreeing linear order*, provided $A(e) <_L v <_L B(e)$, for every $e \in E(H)$ and $v \in e \setminus \{A(e), B(e)\}$. We prove Theorem 3 in the following form.

Theorem 14 *For $r \geq 3$ an r-uniform min&max-marked clique H with $n \geq 2r-2$ vertices has an agreeing linear order if and only if there is an agreeing linear order on every $(2r-2)$-element subset of $V(H)$. Furthermore, the number $2r-2$ in the statement cannot be lowered.*

PROOF: For the second statement we adopt the construction in Theorem 10 to present a min&max-marked clique H on $n = 2r-2$ vertices such that the vertex set of each subhypergraph of H with $2r-3$ vertices has an agreeing linear order but H does not.

Let $V(H) = \{v_1, v_2, \ldots, v_{2r-2}\}$; for $e_0 = \{v_1, \ldots, v_{r-1}, v_r\}$ define $A(e_0) = v_1, B(e_0) = v_{r-1}$, and for every $e = \{v_{i_1}, v_{i_2}, \ldots, v_{i_r}\}$, $1 \leq i_1 < i_2 < \ldots i_r \leq 2r-2$, different from e_0 define $A(e) = v_{i_1}, B(e) = v_{i_r}$. Let $e_1 = \{v_{r-1}, v_r, \ldots, v_{2r-2}\}$.

In an agreeing linear order L of $V(H)$ we have $v_r <_L B(e_0) = v_{r-1} = A(e_1) <_L v_r$, a contradiction. Observe next that $L_1 = (v_1, \ldots, v_{r-1}, v_r, \ldots, v_{2r-2})$ agrees with all edges of $H - e_0$, and $L_0 = (v_1, \ldots, v_r, v_{r-1}, \ldots, v_{2r-2})$ agrees with all edges of $H - e_1$. Because $e_0 \cup e_1 = V(H)$, no subhypergraph $H - v_i$ contains both e_0 and e_1. Therefore, either $L_0 - v_i$ or $L_1 - v_i$ is an agreeing linear order of $V(H) \setminus \{v_i\}$, for every $1 \leq i \leq 2r-2$.

To prove the first part of the theorem define a directed graph G on $V = V(H)$ with an edge (x, y) from x to y if either $x = A(e)$ or $y = B(e)$, for some $e \in E(H)$, $x, y \in e$. Now H has an agreeing linear order if and only if the vertices of G have a labeling $v_1, v_2, \ldots, v_n$ such that each arc (v_i, v_j) of G implies $i < j$. This labeling of V exists if and only if G has no directed cycle. Assume now that H satisfies the Helly-condition. Observe that G has no directed 2-cycle, because if it exists and is induced by $e, f \in E(H)$, then $|e \cup f| \leq 2r-2$, contradicting the condition that on every $(2r-2)$-element subset of $V(H)$ there is an agreeing linear order.

Assume to the contrary that G contains a directed cycle, let $C = (a_1, a_2, \ldots, a_k)$ be a shortest directed cycle of G, $k \geq 3$.

Let the arc (a_1, a_2) of G be induced by some $e \in E(H)$; w.l.o.g. we assume that $a_1 = A(e)$. A shortest directed cycle has no chord, thus $e \cap C = \{a_1, a_2\}$. For the same reason, C contains no r-tuple, hence $k \leq r-1$ and $|e \cup C| = (r+k) - 2 \geq r+2$.

Let $f \subset e \cup C$ be an r-tuple containing C and let $a = A(f)$. If $a \in e \setminus C$, then (a, a_1) is a directed 2-cycle; if $a = a_i$, $i = 1, \ldots, k$, then (a, a_{i-1}) is a directed 2-cycle (with $a_0 = a_k$), a contradiction. □

In addition to the direct proof of Theorem 3 (see Theorem 14 above) we show how Theorem 3 follows as a corollary of Theorem 1. PROOF:[Second proof of Theorem 14]

Let H be an r-uniform min&max-marked clique with at least $2r-2$ vertices, and assume that there is an agreeing linear order on every $(2r-2)$-element subset of $V(H)$. Construct H', a 2-extreme marked clique from H by making the marked vertices undistinguished, and apply Theorem 1 to get an agreeing linear order L, in the sense of Theorem 1.

Every edge $e \in E(H)$ has the property that the smallest and the largest vertex of e in L form the set $\{A(e), B(e)\}$. Call e "good", if the smallest vertex of e in L is $A(e)$, and the largest vertex is $B(e)$, and call it "bad" if it is the other way around. If every edge is good, then L is an agreeing linear order in the sense of Theorem 14, so we are done. If every edge is bad, then the dual of L will work, and we are done. So we just have to settle the case when there is a good edge g, and a bad edge b.

Recall that the Johnson graph is defined on the r-element subsets of $\{1, \ldots, n\}$, as vertices, and two of these vertices (r-sets) e and f are adjacent, if $|e \cap f| = r-1$. An immediate observation is that the Johnson graph is connected.

On the path in the Johnson graph from the good edge g to the bad edge b (which are vertices of the Johnson graph), there are two elements $e, f \in E(H)$ such that $|e \cap f| = r-1$, and e is good and f is bad. Since $|e \cup f| = r+1 \leq 2r-2$, there is an agreeing linear order L' on $e \cup f$.

Note that $|\{A(e), B(e)\} \cap \{A(f), B(f)\}| \geq 1$. Without loss of generality, either $A(e) = B(f)$, or $A(e) = A(f)$. In the former case

$$A(f) <_{L'} B(f) = A(e) <_{L'} B(e),$$

while in the latter,

$$B(f) <_L A(f) = A(e) <_L B(e).$$

In both cases, every element of $e \cap f$ should be both between $A(f)$ and $B(f)$, and $A(e)$ and $B(e)$, contradiction.

□

5 Concluding remarks

In this paper we established Helly-type results for the existence of an agreeing linear order, for *complete* r-uniform hypergraphs, in four versions. For 2-extreme marked, min-marked and min&max marked cliques we found Helly-type theorems and the Helly numbers. For 1-extreme marked cliques we showed that there is no such theorem.

As a possible generalization of the questions discussed in the paper, D. Pálvölgyi (personal communication) proposed the question of investigating Helly-type properties of cliques, such that a poset is specified for each edge as "marks". An edge e agrees with a linear order L, if $L|_e$ is a linear extension of the poset corresponding to e, and an agreeing linear order, as before, is a linear order on the vertex set that agrees with every edge.

Similarly to the proof of Theorem 14, it can be shown that if none of the posets are antichains, and every set of $2r^2$ vertices has an agreeing order, then there is an agreeing linear order of the vertices. However, determining the Helly-numbers (based on the posets) is wide open.

Thinking of 2-extreme marked 3-uniform hypergraphs, it is a basic assumption (see [6, 7]) defining the betweenness relation for all triples with prescribed 'boundaries'. However, it is natural to ask, what is the situation for not necessarily complete hypergraphs. It is not hard to see that in this general case there is no Helly-type theorem in any of the four versions. $u_1, \dots, u_m$ cannot be ordered in a desired way. But then, it is an interesting problem to find conditions on the original hypergraph that would still guarantee a Helly-type theorem. In the 2-extreme marked, min-marked and min&max marked cases, is it enough to assume that the original r-uniform hypergraph is dense (that is, the hypergraph has $\Omega(n^r)$ hyperedges)?

References

[1] R.P. Anstee, A survey of forbidden configuration results, in: Dynamic Surveys, Electron. J. Comb. (2013).

[2] S. Azimipour and P. Naumov, Axiomatic theory of betweenness. arXiv:1902.00847v2

[3] C. Biró, J. Lehel and G. Tóth, Betweenness of convex bodies in the plane. In preparation.

[4] P.C. Fishburn, Betweenness, orders and interval graphs. J. Pure Appl. Algebra 1 (1971), no. 2, 159–178.

[5] Z. Füredi, J. Tao, A. Kostochka, D. Mubayi, and J. Verstraëte, Extremal problems for convex geometric hypergraphs and ordered hypergraphs. Canadian Journal of Mathematics 73, no. 6 (2021): 1648–1666.

[6] E.V. Huntington, A new set of postulates for betweenness, with proof of complete independence. Trans. Amer. Math. Soc. 26 (1924), no. 2, 257–282.

[7] E.V. Huntington and J.R. Kline, Sets of independent postulates for betweenness. Trans. Amer. Math. Soc. 18 (1917), no. 3, 301–325.

Degrees of interior polynomials and parking function enumerators

Tamás Kálmán[1]

Department of Mathematics, Tokyo Institute of Technology and International Institute for Sustainability with Knotted Chiral Meta Matter, Hiroshima University, Japan
H-214, 2-12-1 Ookayama, Meguro-ku, Tokyo 152-8551, Japan
kalman@math.titech.ac.jp

Lilla Tóthmérész [2]

Eötvös Loránd University and
ELKH-ELTE Egerváry Research Group
1117, Pázmány Péter sétány 1/C, Budapest, Hungary
lilla.tothmeresz@ttk.elte.hu

Abstract: The interior polynomial, associated to a directed graph via Ehrhart theory, displays several attractive properties. We express its degree for all so-called semi-balanced digraphs in terms of the minimum cardinality of a directed join. We present a natural extension of this result to oriented regular matroids. By duality, this implies a formula for the degree of the parking function enumerator of an Eulerian directed graph. We extend that result to obtain the degree of the parking function enumerator of an arbitrary rooted directed graph in terms of the minimum cardinality of a certain type of feedback arc set.

Keywords: graph polynomial, greedoid, Ehrhart theory, dijoin

1 Introduction

In this paper we compute the degrees of two interrelated (in fact, in an appropriate sense, dual) graph and matroid polynomials. In particular, we show that they can be expressed using common graph/matroid theoretic notions.

The first type of polynomial we deal with is the interior polynomial of a semi-balanced directed graph [10]. (A *semi-balanced digraph* is a directed graph with a layering of the vertices so that each edge goes exactly one level up. Such structures can be thought of as a generalization of hypergraphs.) It is defined as the h^*-vector of the root polytope associated to the digraph; see Section 2 for detailed definitions. The interior polynomial turns out to be a noteworthy graph invariant. It was first defined in [7] (in a somewhat more restricted setting). It satisfies product and recursion formulas, moreover, it generalizes the specialization $T(x, 1)$ of the Tutte polynomial, as well as the greedoid polynomial of a planar Eulerian branching greedoid, see [10, 14]. In this paper we point out that the degree of the interior polynomial also has a meaningful connection to the graph structure.

A cut in a digraph is *directed* if each of its edges points toward the same shore. An edge set in a digraph is called a *directed join*, or *dijoin* for short, if it intersects each directed cut.

Theorem 1. *Let G be a connected semi-balanced digraph. Then the degree of the interior polynomial of G is equal to $|V(G)| - 1 - \nu(G)$, where $\nu(G) = \min\{|K| \mid K \subseteq E \text{ is a dijoin of } G\}$.*

[1]TK was supported by a Japan Society for the Promotion of Science (JSPS) Grant-in-Aid for Scientific Research C (no. 17K05244).

[2]LT was supported by the National Research, Development and Innovation Office of Hungary – NKFIH, grant no. 132488, by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences, and by the ÚNKP-22-5 New National Excellence Program of the Ministry for Innovation and Technology, Hungary. This work was also partially supported by the Counting in Sparse Graphs Lendület Research Group of the Alfréd Rényi Institute of Mathematics.

Recall that by a theorem of Lucchesi and Younger [12], the quantity $\nu(G)$ above is also the maximal number of edge-disjoint directed cuts in G. Also, if the underlying undirected graph of G is 2-edge connected, then $\nu(G)$ is the minimal number of edges whose reversal yields a strongly connected orientation of G [6, Proposition 9.7.1].

Remark 2. *For semi-balanced digraphs with only two layers, that is, bipartite graphs G in which edges are consistently oriented between the partite classes U and W, there is an obvious inequality*

$$\nu(G) \geq \max\{|U|, |W|\}. \tag{1}$$

On the other hand, the theory of the interior polynomial I_G of a semi-balanced digraph G grew out of the first author's notion of the interior polynomial of a hypergraph [7]. Indeed, by a result of Kálmán and Postnikov [8], the interior polynomial of a hypergraph is equivalent to that of a semi-balanced digraph with only two layers. That case already generalizes $T(x, 1)$ (where $T(x, y)$ is the Tutte polynomial) from graphs to hypergraphs. More to the point, the hypergraphical setup easily yields the inequality

$$\deg I_G \leq \min\{|U|, |W|\} - 1 \tag{2}$$

[7, Proposition 6.1]. Now it is easy to see that Theorem 1, in its two-layered special case, is the statement that the discrepancies of (1) *and* (2) *coincide. This realization, made jointly with András Frank, predates Theorem 1 and was subsumed by it later. Regrettably, neither* (1) *nor* (2) *seem to have generalizations to all semi-balanced digraphs.*

We note that interior polynomials of hypergraphs have another, rather different extension, this time to polymatroids instead of semi-balanced digraphs (see [7], as well as [2] for a two-variable version). As far as we know, the connection to h^*-polynomials does not extend to the polymatroid case. We do not touch on polymatroids in this paper.

Extending Theorem 1, we also compute the degree of the interior polynomial of a co-Eulerian regular oriented matroid. (For definitions, see Section 3.) This is a direct generalization of the graphical case, and we get the analogous answer.

Theorem 3. *Let M be a co-Eulerian regular oriented matroid of rank r. Then the degree of the interior polynomial of M is equal to $r - \nu(M)$, where $\nu(M) = \min\{|K| \mid K \text{ is a dijoin of } M\}$.*

The proof of Theorem 1 readily generalizes to give a proof of Theorem 3. Hence, even though the latter theorem is more general, we give the proof for the graph case for the convenience of those readers who are less interested in oriented matroids.

We also address the question of which semi-balanced orientations of a bipartite graph minimize the degree of the interior polynomial. Note here that each semi-balanced graph is necessarily bipartite. In [10], we conjectured that for an undirected bipartite graph, the orientation of Remark 2 (which we also call a *standard orientation*) minimizes the degree of the interior polynomial among semi-balanced orientations of the graph. Here, we prove this conjecture. It remains an open question to characterize all the degree-minimizing semi-balanced orientations. Also, the analogous problem is unresolved for matroids, where we do not even have a candidate for a degree-minimizing orientation.

The other polynomial we deal with is the enumerator of graph parking functions (also commonly called G-parking functions or generalized parking functions) for rooted directed graphs [13]. This polynomial is equivalent to the greedoid polynomial of the branching greedoid of the rooted digraph [5]. The relationship between the two polynomials is such that the degree of the parking function enumerator corresponds to the number of zeros at the end of the greedoid polynomial.

For Eulerian digraphs, the parking function enumerator agrees with the interior polynomial of its cographic (oriented) matroid [14]. Hence we obtain the following corollary of Theorem 3.

Theorem 4. *The degree of the parking function enumerator of a connected Eulerian digraph G (with any root) is equal to $|E(G)| - |V(G)| + 1 - \operatorname{minfas}(G)$, where $\operatorname{minfas}(G)$ denotes the minimal cardinality of a feedback arc set in G. Equivalently, the coefficient of x^i in the greedoid polynomial of the branching greedoid of G (with any root) is zero for $i = 0, \ldots, \operatorname{minfas}(G) - 1$, and nonzero for $x^{\operatorname{minfas}(G)}$.*

We generalize Theorem 4 to obtain a formula for the degree of the parking function enumerator of an arbitrary rooted directed graph. This setting does not correspond to an interior polynomial anymore. For the general theorem, we define the following rooted variant of a feedback arc set:

Definition 5. *Let G be a root connected digraph with root s. We say that a set of edges $F \subset E$ is an s-connected feedback arc set, if $G[E-F]$ is an s-connected acyclic digraph. We denote by* $\mathrm{minfas}(G,s)$ *the minimum cardinality of an s-connected feedback arc set of G.*

Our formula is as follows.

Theorem 6. *Let $G=(V,E)$ be a root-connected digraph with root r. Then the degree of the parking function enumerator of G, rooted at r, is equal to $|E|-|V|+1-\mathrm{minfas}(G,r)$.*

Equivalently, for the greedoid polynomial of the branching greedoid of G rooted at r, the coefficients of $x^0,\dots,x^{k-1}$ are zero, and the coefficient of $x^{\mathrm{minfas}(G,r)}$ is nonzero.

Remark 7. *Björner, Korte and Lovász show that the constant term of the greedoid polynomial of a (root-connected) rooted digraph G is zero if and only if G contains a directed cycle [4, Theorem 6.10]. Theorem 6 strengthens this statement. Indeed, (for an s-root-connected digraph G) we have* $\mathrm{minfas}(G,s)>0$ *if and only if G contains a directed cycle.*

Acknowledgement We are greatful to András Frank for fruitful discussions, and for pointing out to us Theorem 9.6.12 of [6].

2 The interior polynomial of a semi-balanced digraph

2.1 The degree for a given orientation

Let $G=(V,E)$ be a directed graph. To an edge $e=\overrightarrow{th}\in E$, let us associate the vector $\mathbf{x}_e=\mathbf{1}_h-\mathbf{1}_t\in\mathbb{R}^V$. (Here $t,h\in V$ and $\mathbf{1}_t,\mathbf{1}_h\in\mathbb{R}^V$ are the corresponding generators.)

Definition 8. *The* root polytope *of a directed graph $G=(V,E)$ is the convex hull*

$$\mathcal{Q}_G=\mathrm{Conv}\{\,\mathbf{x}_e\mid e\in E\,\}\subset\mathbb{R}^V.$$

If G is connected, then the dimension of this polytope is either $|V|-1$ or $|V|-2$. The dimension is $|V|-2$ if and only if G satisfies the following condition [10].

Definition 9 (Semi-balanced digraph)**.** *In a directed graph, we say that a cycle is* semi-balanced *if it has the same number of edges going in the two directions around the cycle. We call a directed graph* semi-balanced *if all of its cycles are semi-balanced.*

Beside the above definition, we will also use another characterization of semi-balanced digraphs.

Proposition 10. *[10] A directed graph G is semi-balanced if and only if there is a function $\ell\colon V\to\mathbb{Z}$ such that we have $\ell(h)-\ell(t)=1$ for each edge $\overrightarrow{th}$ of G.*

When it exists, we call the function ℓ above a *layering* of G. In this section, we will examine root polytopes of semi-balanced digraphs. In particular, we will focus on their h^*-polynomials. Let us first recall the definition of this notion.

For any d-dimensional polytope $Q\subset\mathbb{R}^n$ with vertices in $\mathbb{Z}^n$, its *h^*-polynomial* $\sum_{i=0}^d h_i^*t^i$, also commonly called its *h^*-vector*, is defined by Ehrhart's identity

$$\sum_{i=0}^{d}h_i^*t^i=(1-t)^{d+1}\mathrm{Ehr}_Q(t),\quad\text{where}\quad\mathrm{Ehr}_Q(t)=\sum_{k=0}^{\infty}|(k\cdot Q)\cap\mathbb{Z}^n|\,t^k \tag{3}$$

is known as the *Ehrhart series* of Q. We note that $h_0^* = 1$ whenever $d \geq 0$, i.e., whenever Q is non-empty. The h^*-polynomial can be thought of as a refinement of volume. Indeed, $h^*(1)$ is equal to the normalized volume of the polytope, where by normalized we mean that the volume of a d-dimensional unimodular simplex is 1.

Now we are in a position to introduce our object of study for this section.

Definition 11 (Interior polynomial, [7, 8, 10])**.** *For a semi-balanced digraph G, we call the h^*-polynomial of the root polytope $\mathcal{Q}_G$ the* interior polynomial *of G and denote it by I_G.*

Theorem 1. *Let G be a connected semi-balanced digraph. The degree of the interior polynomial of G is equal to $|V(G)| - 1 - \nu(G)$, where*

$$\nu(G) = \min\{|K| \mid K \subseteq E \text{ is a dijoin of } G\}.$$

Now let us start preparing to prove Theorem 1. One ingredient will be the following corollary of Ehrhart–Macdonald reciprocity.

Theorem 12. *[1, Theorem 4.5] Let $P \subset \mathbb{R}^n$ be a d-dimensional ($d \geq 0$) lattice polytope with h^*-polynomial $h_d x^d + \cdots + h_1 x + 1$. Then $h_d = \cdots = h_{k+1} = 0$ and $h_k \neq 0$ if and only if $(d-k+1)P$ is the smallest integer dilate of P that contains a lattice point in its relative interior.*

Notice here that $(d+1)P$ certainly contains an interior lattice point. The degree of the h^*-polynomial of P tells us exactly 'how much sooner' such a point occurs.

In our cases, $\mathcal{Q}_G$ is a $(|V|-2)$-dimensional polytope. Thus if we show that $\nu(G) \cdot \mathcal{Q}_G$ is the smallest integer dilate of the root polytope that contains a lattice point in its interior, then by Theorem 12, it follows that the degree of I_G is indeed $|V| - 1 - \nu(G)$.

For our new quest, a description of $\mathcal{Q}_G$ by half-planes will be useful.

Definition 13. *Let C^* be a cut in the graph G with shores V_0 and V_1. Let f_{C^*} be the functional with $f_{C^*}(\mathbf{1}_v) = 1$ when $v \in V_1$ and $f_{C^*}(\mathbf{1}_v) = 0$ when $v \in V_0$. If G is directed and C^* is a directed cut, we will always suppose that V_1 is the shore containing the heads of the edges in the cut. We will refer to f_{C^*} as the* functional induced by the cut *C^*.*

Let ℓ be a layering of G. We may think of ℓ as a vector (or covector) in $\mathbb{R}^V$. This turns out to be useful for the facet description of the root polytope, that we next give.

Proposition 14. *For any connected semi-balanced graph G with a layering ℓ, we have*

$$\mathcal{Q}_G = \left\{ \mathbf{x} \in \mathbb{R}^V \;\middle|\; \begin{array}{ll} f_{C^*}(\mathbf{x}) \geq 0 & \text{for all elementary directed cuts } C^* \text{ of } G \\ \mathbf{1} \cdot \mathbf{x} = 0 & \\ \ell \cdot \mathbf{x} = 1 & \end{array} \right\}.$$

Here $\mathbf{1} = \sum_{v \in V} \mathbf{1}_v \in \mathbb{R}^V$; the description does not depend on the choice of ℓ because different choices differ by a multiple of $\mathbf{1}$.

Proof. The proof that we give is a direct generalization of the proof of [8, Proposition 3.6], which is a special case of this statement.

It is clear that $\mathbf{1} \cdot \mathbf{x}_e = 0$ and $\ell \cdot \mathbf{x}_e = 1$ hold for each $e \in E(G)$. Similarly, if we let C^* be an elementary directed cut of G, then $e \notin C^*$ clearly implies $f_{C^*}(\mathbf{x}_e) = 0$, whereas when $e \in C^*$, then by the convention of Definition 13, we have $f_{C^*}(\mathbf{x}_e) = 1$. Hence $f_{C^*}(\mathbf{x}_e) \geq 0$ for each $e \in E(G)$. Since each vertex of $\mathcal{Q}_G$ satisfies the conditions of the Proposition, so does the entire root polytope.

Conversely, let $\mathbf{x}$ be a vector that belongs to the right hand side of our formula and consider an arbitrary facet F of $\mathcal{Q}_G$. This needs to contain $|V| - 2$ affine independent vertices, and we fix such a set. The corresponding edges of G form a forest (cf. [10, Proposition 3.1]) of $|V| - 2$ edges, i.e., this is a forest with two connected components. Take the unique elementary cut C^* in the complement of the forest. If C^* was not a directed cut, then there would be edges e both with $f_{C^*}(\mathbf{x}_e) > 0$ and with $f_{C^*}(\mathbf{x}_e) < 0$.

This is not possible because f_{C^*} is linear and F is a facet with $f_{C^*}\big|_F = 0$. Hence C^* is directed, which implies $f_{C^*}(\mathbf{x}) \geq 0$; in particular, $\mathbf{x}$ is on the same side of F as $\mathcal{Q}_G$. Since this holds for every facet, we obtain that $\mathbf{x} \in \mathcal{Q}_G$. □

We saw in the previous proof that each facet of $\mathcal{Q}_G$ is part of a supporting hyperplane that is described by the condition $f_{C^*}(\mathbf{x}) = 0$, where C^* is an elementary directed cut. We may add that whenever C^* is an elementary directed cut, we can select spanning trees of both shores of C^*, thereby obtaining a forest of size $|V| - 2$ with edges from $E - C^*$. The corresponding vectors form a $(|V| - 3)$-dimensional affine independent set that f_{C^*} sends to zero. Hence C^* defines a facet.

Proposition 15. *A point $\mathbf{p} \in \mathcal{Q}_G$ is in the relative interior of $\mathcal{Q}_G$ if and only if there exists a dijoin K of G such that $\mathbf{p} = \sum_{e\in K} \lambda_e \mathbf{x}_e$, where $\lambda_e > 0$ for each $e \in K$ and $\sum_{e\in K} \lambda_e = 1$.*

Proof. By Proposition 14, a point $\mathbf{p} \in \mathcal{Q}_G$ is in the interior of $\mathcal{Q}_G$ if and only if $f_{C^*}(\mathbf{p}) > 0$ for each directed cut C^*. Recall that the functional induced by C^* satisfies $f_{C^*}(\mathbf{x}_e) = 0$ whenever $e \notin C^*$, and $f_{C^*}(\mathbf{x}_e) = 1$ for each $e \in C^*$.

Suppose that $\mathbf{p} = \sum_{e\in K} \lambda_e \mathbf{x}_e$ with $\lambda_e > 0$ for each $e \in K$, where K is a dijoin. Then for any directed cut C^*, we have $f_{C^*}(\mathbf{p}) = \sum_{e\in C^*\cap K} \lambda_e > 0$, since the intersection is nonempty (by the definition of a dijoin), and the summands are all positive. In other words, in this case $\mathbf{p}$ is in the interior of $\mathcal{Q}_G$.

In the other direction, take any convex combination $\mathbf{p} = \sum_{e\in S} \lambda_e \mathbf{x}_e$ with $\lambda_e > 0$ for all $e \in S$. As $\mathcal{Q}_G$ is the convex hull of vectors of the form $\mathbf{x}_e$, we can always find such a formula (usually more than one). We claim that if $\mathbf{p}$ is in the interior, then S is necessarily a dijoin (for any S that arises this way, that is, for any $S \subset E$ so that the convex hull of the corresponding vectors contains an interior point of $\mathcal{Q}_G$). Indeed, suppose that S is disjoint from a directed cut C^*. Then $f_{C^*}(\mathbf{p}) = \sum_{e\in S} \lambda_e \cdot f_{C^*}(\mathbf{x}_e) = 0$, which would contradict $\mathbf{p}$ being an interior point of $\mathcal{Q}_G$. □

The following Lemma is equivalent to the well-known fact that $\mathcal{Q}_F$ is a unimodular simplex for any forest F. See, e.g., [10, Corollary 3.6] for a proof.

Lemma 16. *For a forest F and any positive integer s, a point $\mathbf{p} \in s \cdot \mathcal{Q}_F$ is a lattice point if and only if $\mathbf{p} = \sum_{e\in F} \mu_e \mathbf{x}_e$, where each μ_e is integer.*

Proof of Theorem 1. Let K be a dijoin of G with cardinality $\nu(G)$. Then $\mathbf{p} = \sum_{e\in K} \mathbf{x}_e$ is a point of $\nu(G) \cdot \mathcal{Q}_G$, moreover, it clearly has integer coordinates. Now by Proposition 15 we have that $\mathbf{q} = \frac{1}{\nu(G)}\mathbf{p} = \sum_{e\in K} \frac{1}{\nu(G)} \mathbf{x}_e$ is an interior point of $\mathcal{Q}_G$, which implies that $\mathbf{p}$ is also an interior point of $\nu(G) \cdot \mathcal{Q}_G$.

We also need to prove that for $s < \nu(G)$, there is no interior lattice point in $s \cdot \mathcal{Q}_G$. Suppose that there is an interior lattice point $\mathbf{p} \in s \cdot \mathcal{Q}_G$ for some $s \in \mathbb{Z}_{>0}$ and consider $\mathbf{q} = \frac{1}{s}\mathbf{p}$, which is an interior point of $\mathcal{Q}_G$. Then by Proposition 15 there is a dijoin K such that $\mathbf{q} = \sum_{e\in K} \lambda_e \mathbf{x}_e$, where $\lambda_e > 0$ for each $e \in K$ and $\sum_{e\in K} \lambda_e = 1$. If K contains any cycle, then we can use the basic affine relation associated to the cycle (see, e.g., [10, Lemma 3.3]) to modify the linear combination giving $\mathbf{q}$ so as to make the coefficient of an edge e of the cycle 0. I.e., we obtain $\mathbf{q}$ as a linear combination of elements of $K - e$. Moreover, $K - e$ is also a dijoin because each cut needs to intersect a cycle in at least 2 edges. In conclusion, we may suppose that K contains no cycle, that is, that K is a forest. Now we may apply Lemma 16 to s, K, and $\mathbf{p} = \sum_{e\in K} s\lambda_e \mathbf{x}_e$. This tells us that for $\mathbf{p}$ to be an integer vector, $s\lambda_e$ needs to be an integer for each $e \in K$. Hence altogether, we have $s = \sum_{e\in K} s\lambda_e \geq |K| \geq \nu(G)$. □

2.2 Degree-minimizing orientations of bipartite graphs

Let $\mathcal{G}$ be an undirected bipartite graph with partite classes U and W. In this section, we look at a conjecture from [9] about the degrees of the interior polynomials of different semi-balanced orientations of $\mathcal{G}$.

Any bipartite graph $\mathcal{G}$ has some (typically, many) semi-balanced orientations, but there are two special ones among them: The one where each edge is oriented from U to W, and the one where each edge is oriented from W to U. It is easy to see that these orientations are indeed semi-balanced. We call them

the *standard orientations* of $\mathcal{G}$. The root polytopes of the two standard orientations are isometric, as they are reflections of each other. In particular, their interior polynomials coincide.

In [9] we conjectured that among all semi-balanced orientations of $\mathcal{G}$, the standard orientations minimize every coefficient of the interior polynomial. See [10, Example 6.5] for some concrete instances of this phenomenon. With Theorem 1 in hand, we are able to prove a weakened version of the conjecture.

Theorem 17. *Let $\mathcal{G}$ be a connected, undirected bipartite graph with partite classes U and W. Then among the semi-balanced orientations of $\mathcal{G}$, the degree of the interior polynomial is minimized by the standard orientations.*

Proof. Let G be an arbitrary semi-balanced orientation of $\mathcal{G}$. By the Lucchesi–Younger theorem, the maximal number of disjoint directed cuts in G is equal to $\nu(G)$, the minimal cardinality of a dijoin in G.

As the degree of I_G is equal to $|V(G)|-1-\nu(G)$, the degree of I_G is minimized for those semi-balanced orientations of $\mathcal{G}$ that maximize the number of disjoint directed cuts.

Let $c(\mathcal{G})$ denote the maximal number of disjoint cuts in $\mathcal{G}$. Clearly, for any semi-balanced orientation G of $\mathcal{G}$, we have $\nu(G) \leq c(\mathcal{G})$, since if we take $\nu(G)$ disjoint directed cuts in G, those correspond to disjoint cuts in $\mathcal{G}$.

On the other hand, [6, Theorem 9.6.12] claims that the standard orientations have $c(\mathcal{G})$ disjoint directed cuts. Hence they maximize ν among all orientations (in particular, among all semi-balanced orientations) of $\mathcal{G}$. □

Concrete examples (e.g., [10, Example 6.5]) show that typically, there are some non-standard orientations that also minimize the degree of the interior polynomial. It would be interesting to give a characterization for all the other semi-balanced orientations that attain the minimal degree.

3 A generalization to regular matroids

It turns out that many properties of the root polytopes of digraphs extend word-by-word to regular oriented matroids, moreover, in some applications, one needs this more general case (see, for example, Section 4 or [14]). Here we only sketch how the root polytope and the interior polynomial can be defined for regular oriented matroids, and state the results. The proofs can be obtained by a natural generalization of the arguments of the previous section. We do not properly introduce matroids here. For precise definitions, see for example [3].

From among the many equivalent characterizations of the class of regular matroids, we use the following: A matroid is *regular* if it can be represented by the column vectors of a totally unimodular matrix. Here a matrix is *totally unimodular* if each of its subdeterminants is either 0, -1, or 1. An orientation of a matroid means that the circuits, and the cocircuits are partitioned into positive and negative parts satisfying certain axioms. We denote $C = C^+ \sqcup C^-$, $C^* = (C^*)^+ \sqcup (C^*)^-$. If a matroid is defined using a totally unimodular matrix, then the matrix also yields an orientation.

We call a regular oriented matroid *co-Eulerian* if $|C^+| = |C^-|$ for each circuit C. To a directed graph, one can always associate a regular oriented matroid using its (directed) vertex-edge incidence matrix. The obtained matroid will be co-Eulerian if and only if the orientation of the graph is semi-balanced.

A cocircuit is called *directed* if either $(C^*)^+$ or $(C^*)^-$ is empty. A set K is called a *dijoin* if it intersects each directed cocircuit.

If A is a totally unimodular matrix with columns $\mathbf{a}_1, \ldots, \mathbf{a}_m$, then the *root polytope* $\mathcal{Q}_A$ is defined as $\mathcal{Q}_A = \mathrm{Conv}\{\mathbf{a}_1, \ldots, \mathbf{a}_m\}$. It turns out that if A and A' are two totally unimodular matrices representing the same oriented matroid M, then the h^*-polynomials of $\mathcal{Q}_A$ and $\mathcal{Q}_{A'}$ are the same [14]. Hence this h^*-polynomial is an invariant of the regular oriented matroid M, which we call the *interior polynomial* and denote by I_M. Note that the orientation of the matroid matters: if we keep the (unoriented) matroid structure, but change the orientation, then the interior polynomial might change. (This is true even for graphs, cf. [10, Example 6.5].)

For the dimension of the root polytope, the situation is analogous to the graph case: The dimension of $\mathcal{Q}_\mathcal{A}$ is the rank of A if the corresponding oriented matroid is not co-Eulerian, and $\dim(\mathcal{Q}_A)$ is one less

than the rank of the matroid if its orientation is co-Eulerian [14]. We will only be interested it the latter case. Then, we have the same formula for the degree of I_M as in the graph case.

Theorem 3. *Let M be a co-Eulerian regular oriented matroid of rank r. Then the degree of I_M is equal to $r - \nu(M)$, where*

$$\nu(M) = \min\{|K| \mid K \text{ is a dijoin of } M\}.$$

The proof proceeds through the same steps as in the graph case. One can also ask the analogous question to Theorem 17.

Problem 18. *Given a regular matroid in which each cocircuit has even size, which co-Eulerian orientation has the interior polynomial of smallest degree? Is there any co-Eulerian orientation whose interior polynomial is coordinatewise minimal among the co-Eulerian orientations?*

4 Parking function enumerators and greedoid polynomials

In [14] it is proved that the parking function enumerator of an Eulerian digraph can be expressed as the interior polynomial of the cographic matroid. (Previously, [10] settled the planar case.) Hence the results of the previous section give us information on the degree of the parking function enumerator of an Eulerian digraph, and on the number of zeros at the end of the greedoid polynomial. It turns out, however, that similar results hold for all directed graphs.

In this section we recall the definition of greedoids and the relationship between parking function enumerators and interior polynomials, then we show how to generalize the result on the degree of the parking function enumerator to all directed graphs.

4.1 Preliminaries on greedoids and parking functions

Greedoids were introduced by Korte and Lovász as a generalization of matroids where the greedy algorithm works.

Definition 19 (Greedoid [11])**.** *A set system $\mathcal{F}$ on a finite ground set E is called a greedoid if it satisfies the following axioms*

(1) $\emptyset \in \mathcal{F}$,

(2) for all $X \in \mathcal{F} - \{\emptyset\}$ there exists $x \in X$ such that $X - x \in \mathcal{F}$,

(3) if $X, Y \in \mathcal{F}$ and $|X| = |Y| + 1$, then there exist an $x \in X - Y$ such that $Y \cup x \in \mathcal{F}$.

Elements of $\mathcal{F}$ are called accessible sets, and maximal accessible sets are called bases.

For example, matroids are a special class of greedoids, but greedoids are able to express connectivity properties that matroids cannot. It follows from the axioms that bases have the same cardinality r, which is called the rank of the greedoid.

An interesting subclass of greedoids is the class of directed branching greedoids: For a digraph G, the branching greedoid of G rooted at s is the set system consisting of arborescences of G rooted at s. The bases of the greedoid are the maximal arborescences. (It is easy to check that this is indeed a greedoid.)

The greedoid polynomial was introduced by Björner, Korte and Lovász [4], and it can be defined in many ways (see [4]). Here we recall the definition using activities with respect to a fixed ordering of the edges.

For a basis $B \in \mathcal{F}$ of a greedoid, an ordering $B = \{b_1, \ldots, b_r\}$ is called *feasible* if $\{b_1, \ldots, b_i\} \in \mathcal{F}$ for each $i = 1, \ldots, r$. Note that the axioms guarantee the existence of at least one feasible ordering for each basis. Let us fix an ordering of the groundset E. Now for any basis B of the greedoid, one can associate the lexicographically minimal feasible ordering.

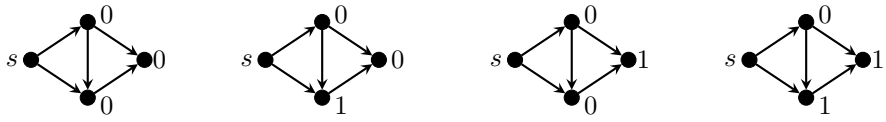


Figure 1: A rooted digraph, and its set of parking functions.

Definition 20 (External activity for greedoids [4]). *Fix an ordering of E. For a basis B, an element $e \notin B$ is externally active in B if for any $f \in B$ such that $B - f + e \in \mathcal{F}$, the lexicographically minimal feasible ordering of B is lexicographically smaller than the lexicographically minimal feasible ordering of $B - f + e$. The external activity of a basis B is the number of externally active elements in B, and it is denoted by $e(B)$.*

Definition 21 (Greedoid polynomial, [4]).

$$\lambda(t) = \sum_{B \text{ basis}} t^{e(B)}$$

We note that this is indeed well-defined, that is, independent of the ordering of the edges used to define the activities.

Swee Hong Chan proves [5] that the greedoid polynomial of a branching greedoid is a simple transformation of the enumerator of graph parking functions. Let us recall these notions, too.

Definition 22 (Graph parking function). *For a directed graph G and a fixed root vertex s, a graph parking function rooted at s is a function $p \in \mathbb{Z}_{\geq 0}^{V-s}$ such that for each $S \subseteq V - s$, there is at least one vertex $u \in S$ with $p(u) < d(V - S, u)$, where $d(V - S, u)$ denotes the number of directed edges leading from $V - S$ to u. Let us denote the set of these functions by $\mathrm{Park}(G, s)$. For a parking function $p \in \mathrm{Park}(G, s)$, let us put $|p| = \sum_{v \in V-s} p(v)$.*

Definition 23 (Parking function enumerator). *For a directed graph $G = (V, E)$ and a fixed root vertex s, the parking function enumerator is the polynomial*

$$\mathrm{park}_{G,s}(x) = \sum_{p \in \mathrm{Park}(G,s)} x^{|p|}.$$

Example 24. *Figure 4.1 shows a rooted digraph and each one of its parking functions. Altogether, the parking funtion enumerator is $x^2 + 2x + 1$.*

The relationship of the greedoid polynomial and the parking function enumerator is the following.

Theorem 25. *[5, Theorem 1.3]* $\lambda_{G,s}(x) = x^{|E|-|V|+1}\mathrm{park}_{G,s}(x^{-1})$

We have previously observed the following connection.

Theorem 26. *[14] Let G be a connected Eulerian digraph, s an arbitrary vertex, and let M be the directed dual matroid of G. Then*

$$\lambda_{G,s}(x) = x^{|E(G)|-|V(G)|+1} I_M(x^{-1}) \qquad \text{and} \qquad \mathrm{park}_{G,s}(x) = I_M(x).$$

Hence if G is Eulerian, we can use Theorem 3 to obtain a formula for the degree of the parking enumerator, or equivalently, a formula for the number of zeros at the end of the greedoid polynomial.

Proof of Theorem 4. For the (directed) cographic matroid M of G, a dijoin of M (that is, a set of edges intersecting each directed cocircuit) corresponds to an edge set of G that intersects each directed cycle. Hence dijoins of M correspond to feedback arc sets of G. Now Theorems 26 and 3 imply the statement of Theorem 4. □

Next, we show how to generalize Theorem 4 to any rooted digraph. For this, we first give a formula for the number of zeros at the end of the greedoid polynomial for an arbitrary greedoid.

4.2 General greedoids

Let us make two easy observations, whose (simple) proofs we leave to the readers.

Definition 27. *Let $X = (E, \mathcal{F})$ be a greedoid, and let $S \subseteq E$. We define $X|_S$ as the pair $(S, \mathcal{F}|_S)$ where $\mathcal{F}|_S = \{A \in \mathcal{F} \mid A \subseteq S\}$, and call it the restriction of X to S.*

Claim 28. *$X|_S$ is a greedoid.*

Claim 29. *Let $X = (E, \mathcal{F})$ be a greedoid and $S \subseteq E$. Fix an ordering of the elements of E, and its restriction to S. Suppose that F is a basis of both X and $X|_S$. An element $e \notin F$ is externally active in F for $X|_S$ (for the above mentioned ordering) if and only if e is externally active in F for X.*

Theorem 30. *Let $X = \{E, \mathcal{F}\}$ be a greedoid of rank r and define $k = \min\{|S| \mid \operatorname{rank}(X|_{E-S}) = r \text{ and } \lambda_{X|_{E-S}}(0) \neq 0\}$. Then for the greedoid polynomial of X, the coefficient of x^i is zero for $i = 0, \ldots, k-1$, and the coefficient of x^k is nonzero.*

Proof. Take an arbitrary basis B of X, and let P be the set of elements of $E - B$ that are externally passive for B. We claim that $B \cup P$ is a set such that $\operatorname{rank}(X|_{B\cup P}) = r$ and $\lambda_{X|_{B\cup B}}(0) \neq 0$. The claim $\operatorname{rank}(X|_{B\cup P}) = r$ follows immediately since B is a basis of X. On the other hand, since elements of P were all externally passive for B in X, this remains so for $X|_{B\cup P}$, thus, in $X|_{B\cup P}$ there are no externally active elements for B. Hence $k \leq |E - B - P|$. As the external activity of B is $e_X(B) = |E - B - P|$, this shows that if the coefficient of x^i is positive, then $i \geq k$.

It remains to show that the coefficient of x^k is positive. Take a set $S \subseteq E$ with $|S| = k$ such that $\operatorname{rank}(X|_{E-S}) = r$ and $\lambda_{X|E-S}(0) \neq 0$. We show that there exist a basis B such that $e_X(B) \leq |S|$. As we already proved that we cannot have $e_X(B) < k$, this will finish the proof.

As $\operatorname{rank}(X|_{E-S}) = r$, each basis of $X|_{E-S}$ is a basis of X. Since $\lambda_{X|_{E-S}}(0) \neq 0$, the greedoid $X|_{E-S}$ has a basis B such that $e_{X|_{E-S}}(B) = 0$. That is, all elements of $E - S - B$ are externally passive in B. Hence, in X, only the elements of S can be externally active for B, thus indeed, $e_X(B) \leq |S| = k$. □

4.3 Arbitrary directed graphs

In this section we use the result of the previous section to generalize Theorem 4 to arbitrary digraphs. Let G be a digraph, and s be an arbitrary fixed vertex of G.

First, note that we can suppose that G is root connected, that is, that each vertex of G is reachable on a directed path from s. (This property is equivalent to G having a spanning arborescence rooted at s.) Indeed, if G is not root connected, then the bases for the branching greedoid of G rooted at s will be the same as the bases for the branching greedoid of G' rooted at s, where G' is the subgraph of G spanned by vertices reachable on a directed path from s. The edges of $G - G'$ will be externally active for any basis, and any edge ordering, hence in this case, $\operatorname{park}_G(x) = \operatorname{park}_{G'}(x)$, and $\lambda_G(x) = x^{|E(G)|-|E(G')|}\lambda_{G'}(x)$.

Theorem 6. *Let $G = (V, E)$ be a root connected digraph. The degree of the parking function enumerator of G rooted at s is equal to $|E| - |V| + 1 - \operatorname{minfas}(G, s)$.*

Equivalently, for the greedoid polynomial of the branching greedoid of G rooted at s, the coefficients of $x^0, \ldots, x^{k-1}$ are zero, and the coefficient of x^k is nonzero for $k = \operatorname{minfas}(G, s)$.

Recall that $\operatorname{minfas}(G, s)$ is the minimal cardinality of an edge set whose removal leaves a root-connected acyclic digraph (Definition 5). It follows from Theorems 4 and 6 that for a connected Eulerian digraph and an arbitrary vertex s, $\operatorname{minfas}(G) = \operatorname{minfas}(G, s)$. It is also not very hard to prove this directly. For general digraphs, $\operatorname{minfas}(G)$ and $\operatorname{minfas}(G, s)$ might differ. For example if G has two vertices, s and v, with one edge from s to v, and two edges from v to s, then $\operatorname{minfas}(G) = 1$ but $\operatorname{minfas}(G, s) = 2$.

Remark 31. *For the interior polynomial (consequently, also for the parking function enumerator of Eulerian digraphs), we had a definition using Ehrhart theory. For the parking function enumerator of general digraphs we are unaware of such a definition.*

We build upon the following result of Björner, Korte and Lovász [4].

Theorem 32. *[4] Let G be a root-connected digraph rooted at s. The greedoid polynomial $\lambda_{G,s}$ has nonzero constant term if and only if G is acyclic.*

Proof of Theorem 6. We apply Theorem 30. If G is root-connected, then the rank of the branching greedoid rooted at s is equal to $|V(G)| - 1$. For an edge set S, the rank of the branching greedoid of $G[E - S]$ remains $|V(G)| - 1$ if and only if $G[E - S]$ is root-connected. On the other hand, Theorem 32 tells us that $\lambda_{G[E-S],s}(0) = 0$ is equivalent to $G[E - S]$ being acyclic. Hence the condition of Theorem 30 indeed gives the condition of Theorem 6 for branching greedoids of root-connected digraphs. □

References

[1] Matthias Beck and Sinai Robins. *Computing the continuous discretely*. Undergraduate Texts in Mathematics. Springer, New York, second edition, 2015. Integer-point enumeration in polyhedra, With illustrations by David Austin.

[2] Olivier Bernardi, Tamás Kálmán, and Alexander Postnikov. Universal Tutte polynomial. *Adv. Math.*, 402:Paper No. 108355, 74, 2022.

[3] Anders Björner, Michel Las Vergnas, Bernd Sturmfels, Neil White, and Günter M. Ziegler. *Oriented matroids*, volume 46 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, Cambridge, second edition, 1999.

[4] Anders Björner, Bernhard Korte, and László Lovász. Homotopy properties of greedoids. *Advances in Applied Mathematics*, 6(4):447 – 494, 1985.

[5] Swee Hong Chan. Abelian sandpile model and biggs–merino polynomial for directed graphs. *Journal of Combinatorial Theory, Series A*, 154:145 – 171, 2018.

[6] András Frank. *Connections in combinatorial optimization*, volume 38 of *Oxford Lecture Series in Mathematics and its Applications*. Oxford University Press, Oxford, 2011.

[7] Tamás Kálmán. A version of Tutte's polynomial for hypergraphs. *Adv. Math.*, 244:823–873, 2013.

[8] Tamás Kálmán and Alexander Postnikov. Root polytopes, Tutte polynomials, and a duality theorem for bipartite graphs. *Proc. Lond. Math. Soc. (3)*, 114(3):561–588, 2017.

[9] Tamás Kálmán and Lilla Tóthmérész. Ehrhart theory of symmetric edge polytopes via ribbon structures. *arXiv:2201.10501*, 2022.

[10] Tamás Kálmán and Lilla Tóthmérész. Root polytopes and Jaeger-type dissections for directed graphs. *Mathematika*, 68(4):1176–1220, 2022.

[11] Bernhard Korte and László Lovász. Mathematical structures underlying greedy algorithms. In *Fundamentals of computation theory (Szeged, 1981)*, volume 117 of *Lecture Notes in Comput. Sci.*, pages 205–209. Springer, Berlin-New York, 1981.

[12] Claudio L. Lucchesi and Daniel H. Younger. A minimax theorem for directed graphs. *J. London Math. Soc. (2)*, 17(3):369–374, 1978.

[13] Alexander Postnikov and Boris Shapiro. Trees, parking functions, syzygies, and deformations of monomial ideals. *Trans. Amer. Math. Soc.*, 356(8):3109–3142, 2004.

[14] Lilla Tóthmérész. A geometric proof for the root-independence of the greedoid polynomial of Eulerian branching greedoids. arXiv:2204.12419, 2022.

Faster Algorithm for Enumerating Maximal Sets of Close Line Segments

Balázs Vass

Department of Telecommunications and Media
Informatics, Budapest University of Technology
and Economics, Budapest, Hungary
balazs.vass@tmit.bme.hu

Abstract: In this study, as a generalization of the line segment intersection problem, we tackle the enumeration of those maximal link sets, in which each pair of links are at most $2r$ apart from one another (for some given $r \geq 0$). We call these link sets as maximal circular disk failures. Specifically, we give a formal problem definition, and briefly present some basic observations and key components of a prior algorithm for determining the set of maximal circular disk failures. We also give some practical parameters of the input to better estimate the number and computing time of maximal circular disk failures. Finally, we give an improved algorithm, that, under practical assumptions, has a running time near linear in the number of line segments.

Keywords: computational geometry, line segment intersection, close line segments, telecommunication networks, disaster resilience, shared risk link groups

1 Introduction

In this Section, we first present the line segment intersection problem, then, as its generalization, the problem of enumerating the maximal sets of close line segments. Since our study is partially motivated by the disaster resilience of telecommunication networks, in the paper, sometimes we think of the circular disks (of a given radius r) as *disasters*. Moreover, a set of close line segments is sometimes referred to as a *Shared Rish Link Group* (SRLG).

1.1 Line segment intersection problem

In the case of a set E of m line segments in the Euclidean plane, the task of listing all line segment intersections is called the *line segment intersection problem* [2]. This problem has many practical applications like the task of overlaying multiple maps among others [2]. Trivially, reporting all edge pairs that intersect, can be done in $O(m^2)$. However, many times, the number of intersections k is much less than $\Theta(m^2)$. Thus, the question arises whether there is an input-sensitive algorithm that, in the case of 'few' intersections ($k \ll m$), runs in a sub-squared time, and if so, what is the lowest possible complexity for it. The following proposition given by Chazelle et al. gives an answer to this:

Proposition 1 (Theorem 5 of [1]) *All k pairwise intersections among m segments in the plane can be computed in $O(m \log m + k)$ time. The running time is optimal. The storage requirement is $O(m + k)$. If so desired, the algorithm will compute the vertical map of the set of segments within the same time and space bounds.*

We note that, based on this proposition, the fastest algorithm deciding whether there exists any line segment intersection runs in $\Theta(m \log m)$. A drawback of the algorithm presented in [1] though is that it is relatively complicated. For this, we also mention a former and simpler algorithm called Bentley-Ottmann [2], which also solves the problem. It has the following complexity.

Proposition 2 (Theorem 2.4 of [2]) *All intersection points of E, together with the segments giving the intersection, can be reported in $O((m + I) \log m)$ time and $O(m)$ space, where I is the number of intersection points.*

We note that in this latter Prop. 2, in each intersecting point, an arbitrary number of line segments can intersect each other. Another observation is that the output of Prop. 2 is more succinct compared to the output provided by Prop. 1.

1.2 Maximal sets of close line segments

Informally speaking, in the current study, as a generalization of the line segment intersection problem, we aim to determine those links that are at most $2r$ apart from one another (for some given $r \geq 0$). More concretely, we are interested in listing all the maximal link sets that can be hit by a circular disk of radius r. We call these link sets as *maximal circular disk failures*. Our output can be seen as a generalization of the output of Prop. 2.

In fact, our article [9] focusing on modeling the effect of a regional disaster hitting a communication network already gave a polynomial algorithm for solving this problem, using the following parameters.

ρ_r is the *link density* of the network which is measured as the maximal number of links that could be hit by a circular disk shaped disaster of radius r.

x is the number of link crossings.

x' is the number of link crossings in E', where all the edges are extended by $3\sqrt{2}r$ in both directions.

μ is the square mean of numbers v_e for all $e \in E$, where v_e is the number of $w \in V \cup X$ such that $d(w, e) \leq 3r$.

These parameters are polynomially bounded in m, and considered to be small in the case of real-life communication backbone networks and disasters, that is in the center of [9]. Denoting by n the number of edge end points, the complexity of the algorithm given therein is:

Proposition 3 (Theorem 7 of [9]) *The maximal circular disk failures with a radius of exactly r can be computed in time $O((n + x)(\log n + \mu\rho_r^5) + x' \log n)$ and this is tight in n.*

2 Main result

The main result resented in this paper is eliminating the additive term $(x' \log n)$ from the complexity estimation of Prop. 3. This also means that parameter x' disappears from the estimation. The improved theorem we will prove is the following.

Theorem 1 *The maximal circular disk failures with radius exactly r can be computed in time $O((n + x)(\log n + \mu\rho_r^5))$ and this is tight in n.*

We note that an intuitive meaning of parameter x' is difficult to capture, at least for telecommunication networks. The elimination of parameter x' from the estimation makes it easier to argue besides the practicality of our algorithm. The key enabler of getting rid of x' is Thm. 5 and related Lemmas.

3 Problem Definition and Basic Results

In the following, based on [9], we give a more thorough formal problem definition, and briefly present some observations and key components of our algorithm for determining the set of maximal circular disk failures. At the end of the paper, as an incremental improvement of former results, we present the proof of Theorems 5 and 1.

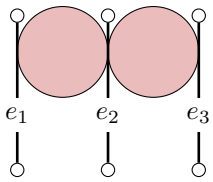


Figure 1: In the figure above, the solid circular disks are disasters with radius r, $d(e_1, e_2) = d(e_2, e_3) = 2r$, while $d(e_1, e_3) = 4r$. The set of regional failures is $F_r = \{\{e_1\}, \{e_2\}, \{e_3\}, \{e_1, e_2\}, \{e_2, e_3\}\}$. The set of maximal regional failures is $M_r = \{\{e_1, e_2\}, \{e_2, e_3\}\}$.

The input is a real number $r \geq 0$ and an undirected connected graph $G = (V, E)$ embedded in the $2D$ plane, where V denotes the set of nodes and E the set of edges (which are also called links). Let $n := |V|$ and $m := |E|$. We assume $n \geq 3$. The edges of G are embedded as line segments, which we call *intervals* in the geometric proofs[1]. A *disk* with center point p *hits* an edge e if its distance to p is at most r.

Definition 4 *A **regional failure** F is a non-empty subset of E, for which there exists a disk with radius r hitting every edge in F.*[2]

Note that the failure of node v is modeled as the failure of all edges incident to node v. Therefore listing the failed nodes beside listing failed edges would not give us additional information from the viewpoint of connectivity.

Definition 5 *Let F_r be the **set of regional failures** of a network for a given radius r.*

According to Def. 4, a subset of a regional failure is also a regional failure. Thus, F_r is a downward closed set minus the empty set.

The network can recover if an SRLG or a subset of links (and nodes) in the SRLG fail simultaneously. In other words, if a regional failure F is listed as an SRLG, then there is no need to list any subset of the links $F' \subsetneq F$ as a new SRLG. The goal is to define a set of SRLGs which covers every possible regional failure and which is of minimal size.

Definition 6 *Let $M_r \subseteq 2^E$ denote the set of SRLGs, for which*

$$M_r \quad = \quad \{F \quad \text{is a regional failure and there is no regional failure } F' \text{ such that } F' \quad \supsetneq \quad F\} \ . \quad (1)$$

In other words, the set of SRLGs M_r is a set of failures caused by disks with radius at most r in which none of the failures is contained in another. Figure 1 illustrates Definitions 4-6. Note that F_r is the set of regional failures, which is the downward closed extension of M_r minus the empty set. A family of sets from the power set of E in which none of the sets is contained in another is called an *antichain* (in the inclusion lattice over 2^E). This antichain is also sometimes called a Sperner system, independent system or a clutter. Note that, M_r is an antichain. Due to the minimality of SRLGs, the following holds.

Proposition 7 *For each SRLG $F \in M_r$, $F \subseteq E$, there is a circular disk c of radius r such that F is exactly the set of edges hit by c.*

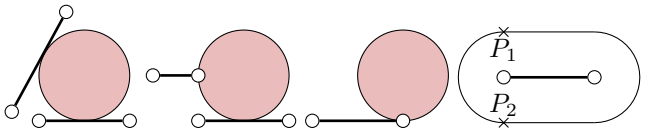
Let r be a tiny positive number. In this case, the list of possible regional failures consists of every *single link or node failure* and link crossings. In other words, this model is a generalization of the 'best practice.' The corresponding antichain can be the set of single node failures, i.e., $|M_r| = n + x$, where x

[1]The case, when edges are considered to be embedded as polygonal chains between their endpoints consisting of at most a constant number of line segments, can also be handled in polynomial time based on the presented results via splitting the polygonal chains up into line segments, running the presented proposed algorithm (sketched in Table 2) for the resulting problem instance, merging the line segments of each polygonal chain, and finally, filtering out the non-maximal sets.

[2]Thus, what we call a regional failure is the worst-case outcome of a disaster damaging an area. F can be seen as a compact representation of all of its subsets.

Table 1: Table of symbols

Notation	Meaning
	General
$G(V, E)$	the network modeled as an undirected connected geometric graph
n, m	number of nodes $\|V\| \geq 3$ and edges $\|E\|$, respectively
r	disaster range ($r \geq 0$)
F	regional failure, i.e.is a non-empty subset of E, for which there exists a disk with radius r hitting every edge in F
F_r	set of regional failures of a network for a given radius r
M_r	F is in M_r if it is is a regional failure and there is no regional failure F' such that $F' \supsetneq F$
c_F	smallest hitting disk of F, where a disk c is smaller than disk c', if c has a smaller radius than c', or if they have equal radius and the center point of c is lexicographically smaller than the center point of c'
X	set of points p which are not in V and there exist at least 2 non-parallel edges crossing each other in p
E_w	$:= \{e \in E \mid d(w, e) \leq 3r\}$; the edges in E_w are in sorted order with respect to the lexicographic ordering of their endpoints
V_e	$:= \{w \in V \cup X \mid d(e, w) \leq 3r\}$
$\mathcal{C}_{r,w}$	The set of the following disks: for $e, f \in E_w$, disks c of radius r (if exist) according to Thm. 2: either case a) applies if e and f are not parallel, and c intersects them in two different points, or case b) when c intersects e and f in two different points, one being an endpoint of e, or case c) when c touches e at an endpoint; moreover we require that formerly computed disks c have centers not farther than $2r$ from w.
$\mathcal{L}_{r,w}$	list of set of edges hit by an element of disk set $\mathcal{C}_{r,w}$
	Parameter
ρ_r	link density of the network, which is measured as the maximal number of links that could be hit by a circular disk shaped disaster of radius r
x	number of link crossings of the network G
μ	square mean of numbers v_e for all $e \in E$, where v_e is the number of $w \in V \cup X$ such that $d(w, e) \leq 3r$
ϕ_r	maximum number of nodes in the $3r$-neighborhood of a link of the input graph G



(a) (b) (c) (d) $N(e, r)$ of an edge e

Figure 2: Case (a),(b) and (c) of Thm. 2 and the neighbourhood $N(e, r)$ of an edge e.

is the number of edge crossings. Informally speaking, protecting node failures is sufficient to protect link failures as well.

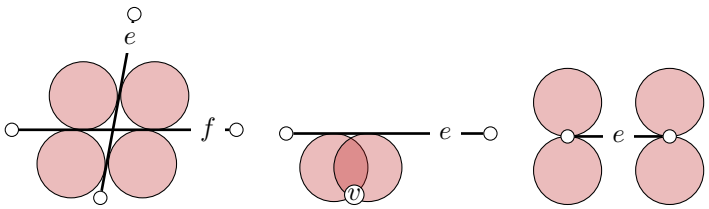
In the following, the aim is to determine the set M_r. At first glance, it is not clear that the cardinality of M_r is 'small.' We will prove polynomial upper bounds on $|M_r|$.

To estimate the size of the SRLG list, let ρ_r denote the maximum number of edges a disk with radius r can hit in the plane, i.e., for every failure F caused by a disk with radius r, $|F| \leq \rho_r$. An observation is that if $\rho_r = O(\log n)$ then there is a polynomial blowup when switching from M_r to F_r, as $|F_r| \leq |M_r| 2^{\rho_r}$. M_r can be treated as a compact representation for F_r. It is also immediate that from F_r one can obtain M_r by $O(|F_r|^2)$ comparisons of subsets of E.

We say a disk c hits a set of edges E_c if it hits all the edges in E_c. Note that several disks can hit the same set of edges.

First, a slight variant of Lemma 9 from [3] is given. This study's assumptions allow somewhat more general topologies with more than 2 collinear points. The segments $e \in H$ are assumed to be nondegenerate.

Theorem 2 (Theorem 1 of [9], see Fig. 2) *Let r be a positive real, and H be a nonempty set of intervals (i.e., edges) from $\mathbb{R}^2$ which is hit by a circular disk of radius r. Then there is a disk c of radius r which hits the intervals of H such that at least one of the following holds (see Fig. 2 for illustrations). (a) There are two non-parallel intervals in H such that c intersects both of them in a single point. These two points are different.*



(a) $\forall e \in E$ and $\forall f \in E$ (b) $\forall e \in E$ and $\forall v \in V$ (c) $\forall e \in E$

Figure 3: The circular disasters examined in Lemma 8

(b) There are two intervals in H such that c intersects both of them in a single point. These two points are different, and one of them is an endpoint of its interval.
(c) Disk c touches the line of an interval $e \in H$ at an endpoint of e.

Lemma 8 *Let H' be a set of intervals from $\mathbb{R}^2$, $|H'| \leq 2$, and r be a positive real number. Then, for every case of Thm. 2, for $H = H'$, a proper circle can be determined in $O(1)$ time.*

Proof: Easy elementary geometric discussion of cases (a), (b) and (c) of Thm. 2. See Fig. 3 for illustration. Note that there can be at most 4 circles that intersect two line segments, as shown in Fig. 3(a), and at most two circles intersecting a line segment and a single point, as shown in Fig. 3(b), and four circles can touch a line at endpoints, as shown in Fig. 3(c). □

4 Algorithm to Enumerate the Set of SRLGs

Next, we define some practical parameters of the input to better estimate the number of SRLGs and computing time. Parameter ρ_r denotes the *link density* of the network, which is measured as the maximal number of links that could be hit by a circular disk shaped disaster of radius r. x is the number of link crossings of the network G. Finally, μ is the square mean of numbers v_e for all $e \in E$, where v_e is the number of $w \in V \cup X$ such that $d(w, e) \leq 3r$. In backbone networks, x is a small number since typically a network node is also installed at each link crossing [5], while the link density ρ_r practically should not depend on the network size. We also know that ρ_r is at least the maximal nodal degree in the graph. For simplicity, we assume that edges intersect in at most one point.

Definition 9 *Let X be the set of points p that are not in V and there exist at least 2 non-parallel edges crossing each other in p. Let $x = |X|$.*

As mentioned above, in backbone network topologies, typically $x \ll n$. This is because a switch is usually installed if two cables are crossing each other[3]. It gives us the intuition that G is "almost" planar, and thus it has few edges.

Claim 10 *The number of edges in G is $\Omega(n)$ and $O(n + x)$.*

Proof: Since G is connected, $m = \Omega(n)$ is immediate. The upper bound was proved in [9] as follows. Let $G'(V \cup X, E')$ be the planar graph obtained from dividing the edges of G at the crossings. Since every crossing increases the number of edges by at least two, $|E'| \geq m + 2x$. On the other hand, $|E'| \leq 3(n + x) - 6$ since G' is planar. Thus $m \leq |E'| - 2x \leq 3n + x - 6$. □

Here we add a note on the Crossing Lemma [6] giving a lower bound on x in function of n and m. For a given graph G, let $\mathrm{cr}(G)$ be the minimum number of edge crossings over the planar embeddings of G. Thm. 6. of [6] states that $\mathrm{cr}(G) \geq \frac{1}{29}\frac{m^3}{n^2} - \frac{35}{29}n$, and if $m \geq 6.95n$, then $\mathrm{cr}(G) \geq \frac{1}{29}\frac{m^3}{n^2}$.

[3]Recent experimental studies give empirical evidence that real-world road networks typically have $\Theta(\sqrt{n})$ edge crossings [4].

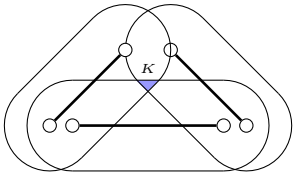


Figure 4: Illustration to Thm. 3

4.1 Lower Bound on Computing the Maximal Failures

Now we present a straightforward lower bound on the time needed to determine M_r. As it will turn out (in Cor. 21), in specific circumstances, this lower bound is asymptotically tight.

Corollary 11 *The complexity of computing M_r is $\Omega(n \log n)$.*

PROOF: By combining Prop. 1 (Lemma 4 of [1]) and Claim 10, we get that reporting that there are no intersecting line segments takes $\Omega(n \log n)$. In other words, this means that computing M_r in the special case of $r = 0$ takes $\Omega(n \log n)$ time. □

4.2 Upper Bounds and Algorithm for Computing the Maximal Failures

The set of link intersections X can be computed in near-linear time, for example, with the help of the Bentley-Ottmann algorithm [2].

Claim 12 *X can be reported in $O((n + x) \log n)$ time and $O(n + x)$ space.*

PROOF: To easily distinguish nodes and edge intersections geometrically, edges are shortened in both directions with a tiny fraction of their length. The statement follows by using Proposition 2 (Theorem 2.4 of [2]) and Claim 10 by also noting that $O(\log(n + x))$ is $O(\log n)$. □

The next theorem states, it is enough to process the edge triplets in the neighborhood with radius $3r$ of every point in $V \cup X$.

Theorem 3 (Thm. 4 of [9]) *For every failure $H \in F_r$ there exists a disk c with radius at most r hitting H with center point at distance at most $2r$ from $V \cup X$.*

Theorem 4 (Theorem 5 of [9]) *Let r be a positive real number, $F \in M_r$ be a set of line segments that can be hit by a disk of radius r. Then there exists a segment $e \in F$ and a disk c described in Thm. 2 (disk c has radius r, hits F, intersects e in a single point Q, and (a), or (b), or (c) holds with $H = F$), such that the center point of c is at distance at most $2r$ from either an endpoint of e or a crossing point (of e and another segment $f \in F$).*

Next, we give better upper bounds on the number of SRLGs. As a consequence of Theorem 4, when considering circular disasters of radius r, then, in a sense, we may ignore the points on the edges $e \in E$ that are more than $3r$ away from $V \cup X$. Consider the pairs (e, v) where $e \in E$, $v \in V \cup X$, and $v \in e$. If we have an SRLG of radius r as in Theorem 4 with edge e such that the distance of c is at most $2r$ from v, then the edges of this SRLG must intersect the disk of radius $3r$ centered at v. This gives at most $15\rho_r$ possibilities for the other edge in addition to e in Theorem 4 (a) or (b) (see Fig. 5, where 15 circular disks of radius r cover a disk of radius $3r$). The number of pairs (e, v) can be counted by looking at the contribution of node v: it will be $\deg(v)$, where deg is the degree in the planarized graph. The sum of the degrees is twice the number of the edges of the latter graph, which is $O(n+x)$. Thus we have the following bound:

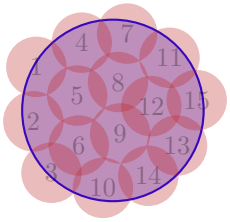


Figure 5: A disk with radius $3r$ can be covered with 15 disks with radius r. Generally, covering a disk with a radius ϵ with the fewest possible number of disks with radii 1 is called the disk covering problem [7].

Corollary 13 $|M_r| = O((n+x)\rho_r)$.

This bound is asymptotically tight (see Sec. IV/A of [9]). Next, we discuss the algorithm to generate the SRLG list.

Table 2: Tasks an respective time complexities of the algorithm for determining M_r and complexity of its tasks

#	Task	Complexity
1	Determine X	$O((n+x)\log n)$
2	For $w \in V \cup X$ determine E_w	$O((n+x)(\log n + \rho_r^2))$
3	For $e \in E$ determine V_e	$O((n+x)(\log n + \rho_r^2))$
4	For $w \in V \cup X$ determine $\mathcal{L}_{r,w}$	$O((n+x)\rho_r^3)$
5	For $e \in E$ for $w_1, w_2 \in V_e$ compare $\mathcal{L}_{w_1}$ with $\mathcal{L}_{w_2}$	$O((n+x)\mu\rho_r^5)$
6	Merge resulting lists in M_r	$O(n+x)$

Theorem 4 together with other formerly presented results inspire an improved algorithm with a running time near linear in n described in Table 2. The main idea is to build up local data structures, precompute the lists of candidate members of M_r, then merge these lists, all in nearly linear time. With this aim, we make the following definitions.

Definition 14 *For a given r and $w \in V \cup X$, let $E_w := \{e \in E |\ d(w,e) \leq 3r\}$; and let the edges in E_w be given in sorted order with respect to the lexicographic ordering of their endpoints. For a given $e \in E$, let $V_e := \{w \in V \cup X |\ d(e,w) \leq 3r\}$.*

Theorem 5 *All sets E_w for $w \in V \cup X$ can be determined in $O((n+x)(\log n + \rho_r^2))$. Similarly, all sets V_e for $e \in E$ can be computed with the same complexity.*

To prove Thm. 5, we need the following three simple lemmas.

Lemma 15 *Let $A = (x,y)$ be a point in the plane of distance at most 3 from the origin. Then, any line going through A intersects either the x- or the y-axis not farther than $3\sqrt{2}$ from the origin.*

Proof: Without loss of generality, we can assume A is in the first quadrant of the plane. Let $B = (0, 3\sqrt{2})$ and $C = (3\sqrt{2}, 0)$, respectively. Then line of BC is tangent to the circle centered at the origin O and having radius 3 (at the point $\left(\frac{3}{\sqrt{2}}, \frac{3}{\sqrt{2}}\right)$, see Fig. 6). Now any line ℓ passing through A must intersect a side of the triangle OBC, hence it intersects at least two sides (Pasch's axiom), therefore ℓ intersects either OB or OC. □

Definition 16 *Let ρ' be the maximum number of edges of E' intersecting a disk with radius $3r$.*

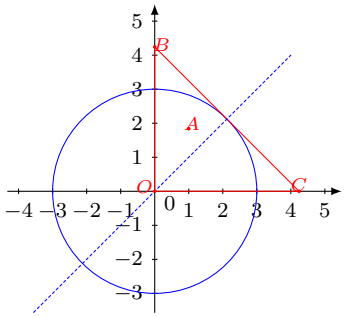


Figure 6: Illustration for proof of Lemma 15

Lemma 17 *ρ' is $O(\rho_r)$.*

Proof: For any point p, the number of edges of G' hit by the disk with radius r and center point p is less or equal to the number of edges of G hit by the disk with radius $(1+3\sqrt{2})r$ and center point p, which is $O(\rho_r)$ since a disk with radius $(1+3\sqrt{2})r$ clearly can be covered by a constant number of disks with radius r. □

Lemma 18 *There are $O((n+x)\rho_r^2)$ intersecting link pairs in link set E' resulting from elongating each edge of E by $3\sqrt{2}r$ in both directions.*

Proof: Let $\{e, f\} \in E$ be two links such that only their elongated versions $\{e', f'\} \in E'$ are crossing in a point z. We claim that this z is on the elongated part of at least one of e' or f', i.e., considering the edges as geometric intervals, $z \in e' \setminus e$ or $z \in f' \setminus f$. Also, for each $e' \in E'$, there are $O(\rho_r)$ edges of E' that cross $e' \setminus e$, since, for an edge $f' \in E'$ to cross $e' \setminus e$, $d(e' \setminus e, f)$ has to be $\leq 3\sqrt{2}r$, and the $3\sqrt{2}r$ neighborhood of $e' \setminus e$ (where $e' \setminus e$ stands of two $3\sqrt{2}r$ long intervals) can be covered with a constant number of disks with radius r. Based on these, and using that $|E'|$ is $O(n+x)$ (Claim 10), we can deduce that there are $O((n+x)\rho_r)$ newly appearing crossing link pairs in E' in addition to those that are crossing in E in a point of $V \cup X$. Regarding to the number of these 'old crossings', in each point of $V \cup X$, there are at most ρ_r links of E crossing (and those links of E' that cross in $V \cup X$, were already counted), meaning $O((n+x)\rho_r^2)$ crossing link pairs. This means a total of $O((n+x)\rho_r^2)$ crossing link pairs in E'. □

With these lemmas in hand, we can present the proof of Thm. 5.

Proof:[Proof of Thm. 5] First, let us concentrate on determining sets E_w for $w \in V \cup X$. Let $G'(V, E')$ be the graph resulting from elongating the edges of E by $3\sqrt{2}r$ in both directions. For reporting link intersections in some slightly modified versions of G', we shall use the Chazelle algorithm [1] that, out of m links, reports all the k intersecting pairs in $O(m \log m + k)$ (Prop. 1).

The most important observation is that, based on Lemma 15, if an edge $e \in E$ is also part of E_w for a $w = (x, y) \in V \cup X$, then the corresponding edge e_{3r} in E' (that was extended in length by $3\sqrt{2}r$ in both directions) intersects either $I_w^{|} := [(x - 3\sqrt{2}r, y), (x + 3\sqrt{2}r, y)]$ or $I_w^{-} := [(x, y - 3\sqrt{2}r), (x, y + 3\sqrt{2}r)]$. Here we use also the simple fact that the diameter of a square (the length of the longest segment within the square) of side length $3r$ is $3\sqrt{2}r$.

Let $G'^{|}$ be the graph resulting by adding intervals $I_w^{|}$ to G' for every $w \in V \cup X$ as edges of the graph. Let $E_w'^{|}$ denote the set of edges (of E') intersecting $I_w^{|}$. $G_w'^{-}$ and $E_w'^{-}$ can be defined similarly. It is easy to see that $E_w'^{|} \cup E_w'^{-}$ contains all the edges, which in the original graph G are not farther from w than $3r$, however, it may contain some outliers. Thus in order to get E_w, one can check the distance of the original (i.e., not extended) edges from w, which correspond to edges in $E_w'^{|} \cup E_w'^{-}$ from w.

It is easy to see that $G'^{|}$ has still $O(n+x)$ edges. We count the number of pairwise intersections in $G'^{|}$ as follows. By Lemma 18, in G', there are $O(n+x)\rho_r^2$ link pairs crossing. In addition to these,

each of the $(n+x)$ new edges (intervals) in $G'^{|}$ intersect $O(\rho_r)$ other edges (since the $3\sqrt{2}r$ neighborhood of each of these $6\sqrt{2}r$ long edges can be covered with a constant number of disks of radius r, and in case of an $e \in E$ elongated as $e' \in E'$, e' crossing a $| \in E'^{|}$ means $d(e,|) \leq 3\sqrt{2}r$). This sums up to $O((n+x)\rho^2 + (n+x)\rho_r)$, that is $O((n+x)\rho_r^2)$. Thus, by Prop. 1, the intersections of $G'^{|}$ can be determined in $O((n+x)\log n + (n+x)\rho_r^2)$, that is $O((n+x)(\log n + \rho_r^2))$ time, alongside with the sets $E'^{|}_w$ for $w \in V \cup X$. The same reasoning applies to the sets E'^{-}_w.

For any given $w \in V \cup X$, $E'^{|}_w \cup E'^{-}_w$ contains $\leq 2\rho'$ edges, this way based on Lemma 17, E_w can be determined in $O(\rho_r \log \rho_r)$ time in such a way that the edges are given in E_w in sorted order with respect to the lexicographic ordering of their endpoints. This means a total complexity of $O((n+x)\rho_r \log \rho_r)$ for this second phase.

The inverse mapping, i.e., sets of nodes V_e for $e \in E$, can be done in the course (or after) the preceding algorithm. Let V_e be initialized as empty set for all edges e, then, when an E_w is confirmed, w is added to sets V_e for all $e \in E_w$. Clearly, this also can be done in the proposed complexity. □

Lemma 19 *The set of SRLGs for circular disk shaped disasters of radius r can be computed in $O((n+x)(\log n + \rho_r^3))$.*

Proof: Based on Claim 12 and Thm. 5, E_w can be determined in the proposed complexity for all $w \in V \cup X$.

Then for every node w, we compute list $\mathcal{L}_{r,w}$ containing the set of edges hit by an element of disk set $\mathcal{C}_{r,w}$ defined as follows: for $e, f \in E_w$ we compute disks c of radius r (if exist) according to Thm. 2: either case a) applies if e and f are not parallel, and c intersects them in two different points, or case b) when c intersects e and f in two different points, one being an endpoint of e, or case c) when c touches e at an endpoint; moreover we require that formerly computed disks c have centers not farther than $2r$ from w. These disks are collected in $\mathcal{C}_{r,w}$. This takes $O((n+x)\rho_r^3)$ time, since there are $O(\rho_r^2)$ disks c to determine and store in $\mathcal{C}_{r,w}$, and for each $c \in \mathcal{C}_{r,w}$ the set of edges hit by c can be determined in $O(\rho_r)$ time based on E_w. It follows readily from Thm. 4 that for every $F \in M_r$ there exists a $w \in V \cup X$ such that F is a subset of an element of list $\mathcal{L}_{r,w}$. □

Please note that lists $\mathcal{L}_{r,w}$ together may contain duplicates and non-maximal sets as well, those will be eliminated later at a subsequent phase.

As mentioned after Lemma 19, the final task for determining M_r is to merge lists $\mathcal{L}_{r,w}$ by eliminating duplicates and non-maximal elements. To do this in subquadratic time in n, one must avoid comparing all pairs of lists $\mathcal{L}_{r,w_1}$, $\mathcal{L}_{r,w_2}$.

Definition 20 *Let μ be the mean square of numbers $|V_e|$ for all $e \in E$, i.e. $\mu := \frac{\sum_{e\in E} |V_e|^2}{m}$.*

Now we can state the proof of Thm. 1.

Proof: According to Lemma 19, all sets of failures $\mathcal{L}_{r,w}$ can be determined in time $O((n+x)(\log n+\rho_r^3))$.

We observe that it is enough to compare lists $\mathcal{L}_{r,w_1}$ and $\mathcal{L}_{r,w_2}$ for possible containment or duplicates only if $E_{w_1} \cap E_{w_2} \neq \emptyset$, or, in other words, there exists an $e \in E$ for which $\{w_1, w_2\} \subseteq V_e$. We deduce that it is enough to compare for all $e \in E$ and $w_1, w_2 \in V_e$ list pairs $\mathcal{L}_{r,w_1}$, $\mathcal{L}_{r,w_2}$. This means comparing at most

$$\sum_{e\in E} \frac{|V_e|(|V_e|-1)}{2} < m\frac{\sum_{e\in E}|V_e|^2}{m} = m\mu \overset{\text{Claim } 10}{=} O((n+x)\mu)$$

pairs of lists, with each list having $O(\rho_r^2)$ elements. Taking into consideration that a comparison of two elements (SRLG candidates) can be done in $O(\rho_r)$, we obtain a complexity of $O((n+x)\mu\rho_r^5)$, confirming the claim for the total complexity. The lower bound is provided by Corollary 11. □

Table 2 summarizes the steps of our proposed algorithm. Note that parameters ρ_r, x, and μ are theoretically upper bounded by m, $\frac{m(m-1)}{2}$, and $(n+x)^2$, respectively, meaning that our algorithm for

determining M_r is clearly polynomial in n or m. Furthermore, based on Thm. 1 using that x is $O(n)$ in practice, and that ρ_r is more or less proportional to $\frac{2r}{diam}m$ ([8])) in the interval $(0, diam/2]$, where $diam$ is the geometric diameter of the network, we get a complexity bound of $O\left(n(\log n + \mu(\frac{r}{diam})^5)\right)$ for determining M_r. Also, as in practice $x = O(n)$, and for r much smaller than network diameter, $\rho_r = O(1)$, and $\mu = \log(n)$ we can state that:

Corollary 21 *If $\rho_r = O(1)$, $\mu = O(\log n)$, and x is $O(n)$, M_r can be calculated in $O(n \log n)$ optimal time. These assumptions hold in practice when r is much smaller than the geographical network diameter.*

Proof: Combining Thm. 1 and Cor. 11 yields the proof. □

Instead of μ, we may use a more intuitive parameter, namely, ϕ_r:

Definition 22 *Parameter ϕ_r denotes the maximum number of nodes in the $3r$-neighborhood of a link of the input graph G.*

Proposition 23 *M_r^p can be determined in $O\left((|V| + x)\left(\log |V| + \phi_r^2 \rho_r^5\right)\right)$.*

5 Conclusion

In this paper, we presented an improved polynomial algorithm for enumerating the maximal link sets that can be simultaneously hit by a circular disk of a given radius $r \geq 0$. This problem can be viewed as a generalization of the line segment intersection problem (where $r = 0$). In certain practical settings, the algorithm proposed in this paper has an optimal near-linear time complexity.

Acknowledgements

I thank Dömötör Pálvölgyi and János Tapolcai for the fruitful discussions on this topic.

This research was partially supported by the National Research, Development and Innovation Fund of Hungary (grant No. 135606). Supported by the ÚNKP-22-4-II-BME-248 New National Excellence Program of the Ministry for Culture and Innovation from the source of the National Research, Development and Innovation Fund.

References

[1] Chazelle, B. & Edelsbrunner, H. An optimal algorithm for intersecting line segments in the plane. *Jurnal Of The ACM (JACM)*. (1992)

[2] Berg, M., Cheong, O., Kreveld, M. & Overmars, M. Computational Geometry: Algorithms and Applications. (Springer Berlin Heidelberg,2008)

[3] Neumayer, S., Zussman, G., Cohen, R. & Modiano, E. Assessing the vulnerability of the fiber infrastructure to disasters. *IEEE/ACM Trans. Netw.*. **19**, 1610-1623 (2011)

[4] Eppstein, D. & Goodrich, M. Studying (non-planar) road networks through an algorithmic lens. *Proceedings Of The 16th ACM SIGSPATIAL International Conference On Advances In Geographic Information Systems*. pp. 1-10 (2008)

[5] Eppstein, D., Goodrich, M. & Strash, D. Linear-Time Algorithms for Geometric Graphs with Sublinearly Many Edge Crossings. *SIAM Journal On Computing*. **39**, 3814-3829 (2010)

[6] Ackerman, E. On topological graphs with at most four crossings per edge. *Computational Geometry*. **85** pp. 101574 (2019), http://www.sciencedirect.com/science/article/pii/S0925772119301154

[7] Kershner, R. The Number of Circles Covering a Set. *American Journal Of Mathematics*. **61**, 665-671 (1939), http://www.jstor.org/stable/2371320

[8] J. Tapolcai, L. Rónyai, B. Vass, and L. Gyimóthi, "List of Shared Risk Link Groups Representing Regional Failures with Limited Size", in Proc. IEEE INFOCOM, Atlanta, GA, USA, 2017

[9] J. Tapolcai, L. Rónyai, B. Vass, and L. Gyimóthi, "Fast Enumeration of Regional Link Failures Caused by Disasters with Limited Size", IEEE-ACM Transactions on Networking, 2020

The importance of being series-parallel

Dedicated to the memory of Professor Takao Nishizeki,
one of the founders of the series of the Japanese-
Hungarian Symposia on Discrete Mathematics and Its Applications

András Recski[1]

Department of Computer Science and Information Theory
Budapest University of Technology and Economics
1111 Budapest Műegyetem rkp. 3., Hungary
recski@cs.bme.hu

Áron Vékássy[1]

Department of Computer Science and Information Theory
Budapest University of Technology and Economics
1111 Budapest Műegyetem rkp. 3., Hungary
aron.vekassy@cs.bme.hu

Abstract: Duffin has characterized series-parallel graphs during his study of 1-ports, his result has been generalized by Nishizeki and Saito for 3-terminal 2-ports. A graph is series-parallel if and only if it does not contain K_4 or its series extension as a subgraph. Electric networks with series-parallel topology play an important role in the qualitatively reliable synthesis of certain devices. In this short note we show that series-parallel graphs arise in network synthesis in another context as well, and the cycle matroid of K_4 turns out to appear in all the known singular network constructions in which these singularities are caused by numerical instabilities.

Keywords: ***n*-ports, series-parallel, singularity**

1 Introduction

One of the main research areas of Professor Takao Nishizeki (1947-2022) was the theory and applications of series-parallel graphs. He and his coauthors had several important results showing that certain graph properties can be recognized in polynomial, sometimes even in linear time if the input is restricted to series-parallel graphs, see [1], [2], [3], [4], [5], [6], [7], [8], [9]. His initial interest in series-parallel graphs was motivated by the classical result of Duffin [10], that the complete graph K_4 on four vertices (a. k. a. a Wheatstone-bridge in the context of electric network theory) is the smallest non-series-parallel graph. In fact, a graph is series-parallel if and only if neither K_4 nor its series extensions are contained in it as a subgraph. One-ports, the simplest building blocks in electric network theory, have two specific vertices, called terminals, and Duffin's result is often formulated as a characterization of two-terminal series-parallel graphs (see [1] for a formal definition). The second simplest building blocks in network theory are the three-terminal two-ports and, accordingly, the results of Nishizeki and Saito [11] generalized Duffin's results for three-terminal graphs.

[1]The research reported in this paper and carried out at the Budapest University of Technology and Economics was supported by the TKP2020, National Challenges Program of the National Research Development and Innovation Office (BME NC TKP2020) and by the Higher Education Excellence Program of the Ministry of Human Capacities in the frame of the Artificial Intelligence research area of the Budapest University of Technology and Economics (BME FIKP-MI/SC).

2 Series-parallel graphs in network synthesis

In electrical network theory, a linear n-port is described by the equation $\mathbf{Au} + \mathbf{Bi} = \mathbf{0}$ where $\mathbf{u}$ and $\mathbf{i}$ correspond to the voltages and the currents of the ports, respectively and both $\mathbf{A}$ and $\mathbf{B}$ have n columns. The matroid $\mathcal{M}$ of an n-port is defined as the column space matroid of its matrix $\mathbf{M} = (\mathbf{A} \mid \mathbf{B})$. The rank of an n-port can be defined as follows: $r = r(\mathbf{M})$. If the equation $n = r$ holds true we call the n-port *ordinary*. Some particular n-ports mentioned in this paper are described in Table 1.

Name	Matrix	Rank
Resistor	$\mathbf{M} = \begin{pmatrix} 1 & -R \end{pmatrix}$	1
Norator	$\mathbf{M} = \begin{pmatrix} 0 & 0 \end{pmatrix}$	0
Nullator	$\mathbf{M} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	2
Ideal transformer	$\mathbf{M} = \begin{pmatrix} -1 & k & 0 & 0 \\ 0 & 0 & k & 1 \end{pmatrix}$	2
Gyrator	$\mathbf{M} = \begin{pmatrix} 0 & -1 & R & 0 \\ 1 & 0 & 0 & R \end{pmatrix}$	2
Nullor	$\mathbf{M} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$	2
3-port circulator	$\mathbf{M} = \begin{pmatrix} 1 & -1 & 0 & 1 & 1 & 0 \\ 0 & 1 & -1 & 0 & 1 & 1 \\ -1 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}$	3
4-port circulator	$\mathbf{M} = \begin{pmatrix} 1 & -1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 1 & 1 \\ -1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}$	4

Table 1: Some important network elements

If the n-port is obtained from the interconnection of other n-ports along the graph G then $\mathcal{M}(\mathbf{M}) = (\mathcal{G} \vee \mathcal{A})/(E^u_{Int} \cup E^i_{Int})$ if the genericity assumption holds where $\mathcal{M}(\mathbf{M})$ is the matroid of the new n-port. $\mathcal{G}$ is the direct sum of the cycle matroid on the set of edges corresponding to currents of G, and the cocycle matroid on the set of edges corresponding to voltages of G. $\mathcal{A}$ is the direct sum of the matroids of the interconnected multiports and E^u_{Int} and E^i_{Int} are the edges corresponding to internal voltages and currents respectively [12]. Note that $\mathcal{G}$ represents the Kirchoff equations obtained from the network topology and $\mathcal{A}$ contains information on the n-ports to be interconnected.

The genericity assumption means that we do not allow cancellations among the nonzero parameters constituting the matrix of the different n-ports to be interconnected. However if we drop this genericity assumption there can be particular choices of parameters which result in cancellations that can alter the resulting matroid of the network. A realization of an n-port by the interconnection of other n-ports is called qualitatively reliable (QR) if the matroid of the resulting networks remains the same no matter how these parameters are chosen [13].

This is perhaps best illustrated through an example of [13]. Consider the networks of Figure 1.

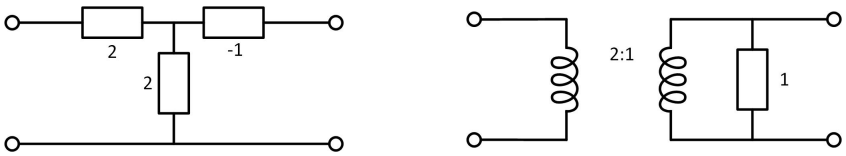


Figure 1: A non-QR and a QR synthesis

They both synthesize the 2-port with the describing matrix $\mathbf{M} = \left(\begin{smallmatrix} -1 & 2 & 0 & 0 \\ 0 & -1 & 2 & 1 \end{smallmatrix}\right)$. Notice that in the general case the 2-port on the right has the matrix $\mathbf{M}' = \left(\begin{smallmatrix} -1 & k & 0 & 0 \\ 0 & -G & k & 1 \end{smallmatrix}\right)$ where k is the transfer ratio of the transformer and G is the conductance of the resistor. Observe that $\mathcal{M}(\mathbf{M}) = \mathcal{M}(\mathbf{M}')$. If, however, we change the resistance values of the network on the left, the last two columns of its describing matrix will not be parallel anymore, therefore only the synthesis on the right is QR.

Electric networks with series-parallel topology are known to play an important role in the QR synthesis of certain devices [13]. For example, using the properties of gammoids one can show that circulators cannot be synthesized in a qualitatively reliable way from one-ports and two-ports interconnected along series-parallel topology.

Recently series-parallel graphs arose in network synthesis in another context: We gave a canonical synthesis of those multiports which are reciprocal and antireciprocal at the same time [14]. For the undefined concepts of electric network theory the reader is referred to [14].

An n-port has a hybrid description if it has rank n and its equations can be written in the form:

$$\begin{bmatrix} \mathbf{u}_1 \\ \mathbf{i}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{C} \\ \mathbf{D} & \mathbf{G} \end{bmatrix} \begin{bmatrix} \mathbf{i}_1 \\ \mathbf{u}_2 \end{bmatrix}$$

where $\mathbf{u}_1$ and $\mathbf{i}_1$ are the voltages and currents of a subset of the ports and $\mathbf{u}_2$ and $\mathbf{i}_2$ are the voltages and currents of the rest of the ports.

A reciprocal n-port always has a hybrid description, and its hybrid matrix looks like this:

$$\mathbf{H} = \begin{bmatrix} \mathbf{R} & \mathbf{C} \\ -\mathbf{C}^T & \mathbf{G} \end{bmatrix}$$

where $\mathbf{R}$ and $\mathbf{G}$ are symmetrical and the parameters within them have dimensions of resistance and conductance, respectively, while the parameters in the matrix $\mathbf{C}$ are without dimension [15]. The sizes of the matrices $\mathbf{R}$, $\mathbf{G}$ and $\mathbf{C}$ are $p \times p$, $q \times q$ and $p \times q$, respectively. If the reciprocal n-port is antireciprocal as well then $\mathbf{R}$ and $\mathbf{G}$ are zero matrices. Based on this observation we show in [14] that all n-ports that are reciprocal and antireciprocal at the same time can be synthesized using ideal transformers only. This synthesis is presented on Figure 2, here the transfer ratios of the ideal transformers are simply the nonzero entries of the matrix $\mathbf{C}$.

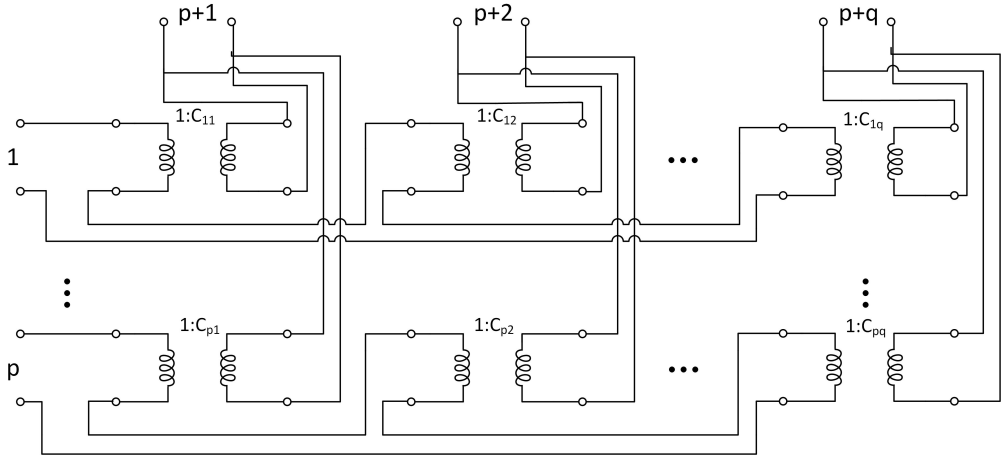


Figure 2: Synthesis construction

3 Singular networks, "realizing" nullators and/or norators

Since our synthesis solution in the previous section needed series-parallel topologies only, it was natural to ask what happens if two ideal transformers are interconnected along the interconnection graph K_4 to form a new 2-port. In the general case this interconnection (Figure 3) leads to an ordinary ideal transformer.

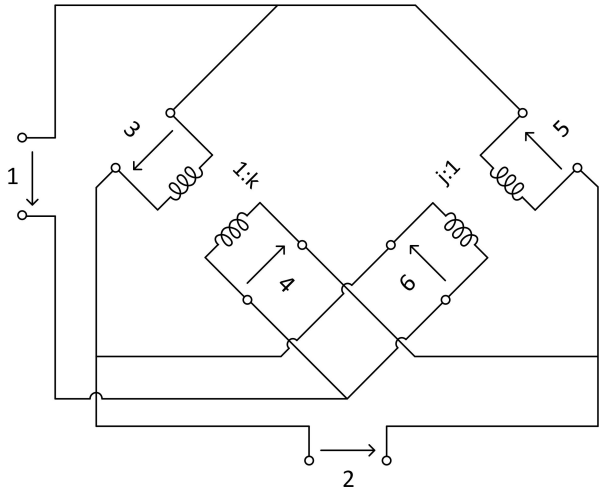


Figure 3: Singular interconnection

We have found in [14] that this network is highly singular if either $k = 1$ and $j = -1$ or $k = -1$ and $j = 1$. In the first case the 2-port becomes a pair of nullators, while in the second it behaves as a pair of norators.

In his article Carlin [16] showed that the synthesis of nullators or norators using ordinary n-ports can only be achieved through such singularities. More specifically, "We [...] expect that any equivalent circuit which represents these elements has infinite sensitivity. That is if some circuit element in the equivalent structure for a nullator or norator is changed slightly, the terminal performance will no longer be similar to that of the nullator or norator.". In other words, he claimed that there is no QR synthesis for these 1-ports.

A small addition to this is that we showed in [17] that one can construct absurd examples (one is shown in Figure 4) that realize a nullor in the general case as well, therefore are QR. However in these networks it is obvious that some "forbidden" interconnections cause the strange behavior as it makes no sense from an engineering point-of-view to connect a current source and an open circuit in series.

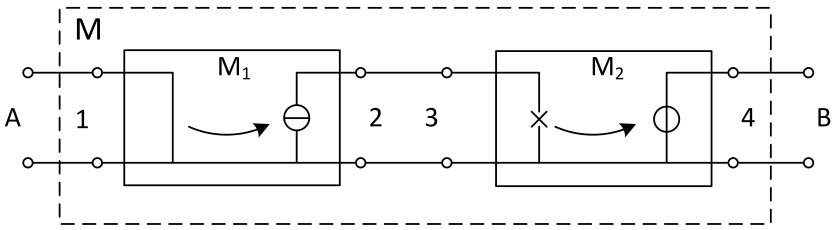


Figure 4: QR synthesis of a nullor

There were some previous examples in the literature for networks "realizing" nullators or norators as a result of some singularities. Such examples (shown on Figure 5) are the gyrator network of Carlin and Youla [18] and the 3-port and 4-port circulator networks of Carlin [16]. Note that all three of these

examples behave like regular networks in the general case and become nullator or norator realizations in two singular cases each. Consequently this kind of realization is not QR and is much more interesting than the QR one, as here it is not clear what exactly causes these networks to show an entirely different behavior at the points of these singularities. Therefore it might be worth to examine the known networks that are singular in this sense, in an attempt to characterize aspects of this behavior.

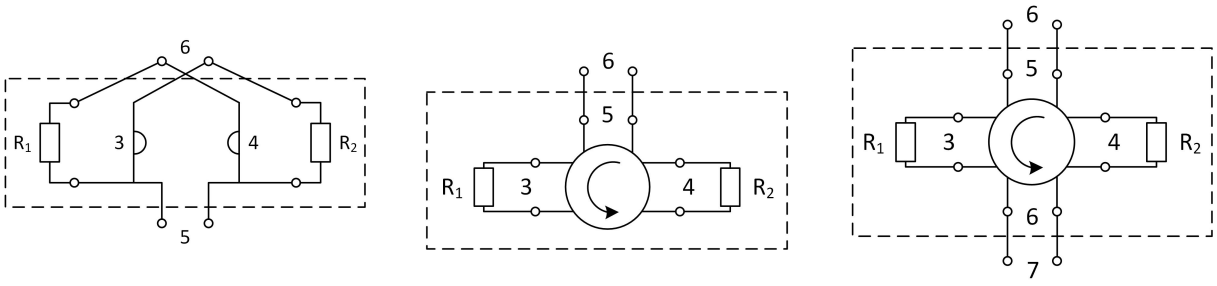


Figure 5: A nullor realization, the 3-port circulator network and the 4-port circulator network

The topology of the first example on Figure 5 is the same K_4 as in our example on Figure 3. While the topology of the two circulator-networks is trivially series-parallel, K_4 still arises in an unexpected way. Recall that the matroid $\mathcal{M}$ describing the qualitative properties of the network arises as a minor of $\mathcal{M} = \mathcal{G} \vee \mathcal{A}$, where $\mathcal{G}$ is the direct sum of the cycle and cocycle matroids of the graph of the interconnection and $\mathcal{A}$ describes the algebraic properties of the devices. The matroid $\mathcal{A}$ of the 3-port circulator is just the cycle matroid of K_4 [19] and that of the 4-port circulator contains the cycle matroid of K_4 as a minor. This is summarized in Table 2.

Does the cycle matroid of K_4 appear...	in $\mathcal{G}$?	in $\mathcal{A}$?
The realization of a nullator-norator pair using a gyrator [18]	Yes	No
The realization of a nullator or a norator using a 3-port circulator [16]	No	Yes
The realization of a nullator-norator pair using a 4-port circulator [16]	No	Yes
The realization of a pair of nullators or a pair of norators [14]	Yes	No

Table 2: Appearence of K_4 in nullator/norator constructions

Hence we can conclude that, in one way or another, the graph K_4 or its cycle matroid appears in all the known singular network constructions which are not QR.

References

[1] K. Takamizawa, T. Nishizeki, and N. Saito. Linear-time computability of combinatorial problems on series-parallel graphs. *J. ACM*, 29(3):623–641, 1982.

[2] X. Zhou, H. Suzuki, and T. Nishizeki. A linear algorithm for edge-coloring series–parallel multigraphs. *Journal of Algorithms*, 20(1):174–201, 1996.

[3] X. Zhou, H. Suzuki, and T. Nishizeki. An NC parallel algorithm for edge-coloring series–parallel multigraphs. *Journal of Algorithms*, 23(2):359–374, 1997.

[4] T. Nishizeki, J. Vygen, and X. Zhou. The edge-disjoint paths problem is NP-complete for series–parallel graphs. *Discrete Applied Mathematics*, 115(1):177–186, 2001.

[5] T. Fujino, X. Zhou, and T. Nishizeki. List edge-colorings of series-parallel graphs. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E86A:1034–1045, 05 2003.

[6] X. Zhou and T. Nishizeki. Multicolorings of series-parallel graphs. *Algorithmica*, 38(2):271–297, 2003.

[7] X. Zhou, Y. Matsuo, and T. Nishizeki. List total colorings of series-parallel graphs. *Journal of Discrete Algorithms*, 3(1):47–60, 2005.

[8] Y. Matsuo, X. Zhou, and T. Nishizeki. Sufficient condition and algorithm for list total colorings of series-parallel graphs. *IEICE Transactions*, 90-A:907–916, 05 2007.

[9] X. Zhou and T. Nishizeki. Orthogonal drawings of series-parallel graphs with minimum bends. *SIAM Journal on Discrete Mathematics*, 22:1570–1604, 01 2008.

[10] R.J Duffin. Topology of series-parallel networks. *Journal of Mathematical Analysis and Applications*, 10(2):303–318, 1965.

[11] T. Nishizeki and N. Saito. Necessary and sufficient condition for a graph to be three-terminal series-parallel. *IEEE Transactions on Circuits and Systems*, 22(8):648–653, 1975.

[12] A. Recski. Contributions to the n-port interconnection problem by means of matroids. In *A. Hajnal and V.T. Sós (eds), Combinatorics, Colloquia Mathematica Societatis János Bolyai*, volume 18, pages 877–892, 1978.

[13] A. Recski. Matroids and network synthesis. In *Proc. European Conf. on Circuit Theory and Design*, pages 192–197, 1980.

[14] A. Recski and Á. Vékássy. Synthesis of a class of reciprocal n-ports and a highly singular construction. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 70(1):3–5, 2023.

[15] L. Chua and Y. Lam. Dimension of N-ports. *IEEE Transactions on Circuits and Systems*, 21(3):412–416, 1974.

[16] H. Carlin. Singular network elements. *IEEE Transactions on Circuit Theory*, 11(1):67–72, 1964.

[17] A. Recski and Á. Vékássy. Interconnection, reciprocity and a hierarchical classification of generalized multiports. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 68(9):3682–3692, 2021.

[18] H. J. Carlin and D. C. Youla. Network synthesis with negative resistors. *Proceedings of the IRE*, 49(5):907–920, 1961.

[19] A. Recski. Unique solvability and order of complexity of linear networks containing memoryless n-ports. *Circuit Theory Appl.*, 7:31–42, 1979.

Fault-tolerance of leaf-guaranteed graphs

Jan Goedgebeur[1]

Department of Computer Science
KU Leuven Kulak
Etienne Sabbelaan 53, 8500 Kortrijk, Belgium
and
Department of Applied Mathematics, Computer Science and Statistics
Ghent University
Krijgslaan 281-S9, 9000 Ghent, Belgium
jan.goedgebeur@kuleuven.be

Jarne Renders[1]

Department of Computer Science
KU Leuven Kulak
Etienne Sabbelaan 53, 8500 Kortrijk, Belgium
jarne.renders@kuleuven.be

Gábor Wiener[2]

Department of Computer Science and Information Theory
Budapest University of Technology and Economics
Műegyetem rkp. 3., 1111 Budapest, Hungary
wiener@cs.bme.hu

Carol T. Zamfirescu

Department of Applied Mathematics, Computer Science and Statistics
Ghent University
Krijgslaan 281-S9, 9000 Ghent, Belgium
and
Department of Mathematics
Babeş-Bolyai University
Cluj-Napoca, Roumania
czamfirescu@gmail.com

Abstract: We study fault-tolerance of networks using the minimum leaf number of the corresponding graph G (that is the minimum number of leaves of the spanning trees of G) and its vertex-deleted subgraphs. Our main notion is the so-called *fault cost*, which is based on the number of vertices that have different degrees in minimum leaf spanning trees of G and its vertex-deleted subgraphs.

Keywords: hamiltonicity, traceability, spanning tree, minimum leaf number

1 Introduction

This is a shortened, conference version of the paper, containing none of the proofs. We investigate the fault-tolerance of networks using spanning trees of the corresponding graphs. Optimisation problems concerning spanning trees occur in various applications, such as querying in computer database systems and connection routing. Throughout the paper, we assume graphs to be undirected and 2-connected, unless explicitly stated otherwise. The vertex set and the edge set of a graph G is denoted by $V(G)$ and $E(G)$, respectively. The subgraph of G induced by $X \subseteq V(G)$ is denoted by $G[X]$ and let $G - X := G[V(G) \setminus X]$, $G - v := G - \{v\}$ for any $v \in V(G)$. For $a, b \in V(G)$ let $G + (a, b)$ denote the graph obtained from G by adding (a, b) to $E(G)$. The set of all spanning trees of G is denoted be $\mathcal{T}(G)$. Denote by $L(T)$

[1]Research is supported by Internal Funds of KU Leuven

[2]Research is supported by project no. BME-NVA-02, implemented with the support provided by the Ministry of Innovation and Technology of Hungary from the National Research, Development and Innovation Fund, financed under the TKP2021 funding scheme.

the set of leaves of a tree T and put $\ell(T) := |L(T)|$. Following [9], the *minimum leaf number* $\mathrm{ml}(G)$ of G is defined to be 1 if G is hamiltonian and $\min_{T \in \mathcal{T}(G)} \ell(T)$ otherwise. For further results on trees with a minimum number of leaves and related problems, we refer to [3, 8, 1, 5].

A graph G is said to be *k-leaf-guaranteed* if $k = \mathrm{ml}(G) \geq \mathrm{ml}(G - v)$ for all $v \in V(G)$. We denote the family of all k-leaf-guaranteed graphs with $\mathcal{L}_k$ and call a graph $G \in \bigcup_k \mathcal{L}_k$ *leaf-guaranteed*. It is easy to show that $\{\mathrm{ml}(G - v)\}_{v \in V(G)} \subset \{k - 1, k\}$ for any k-leaf-guaranteed graph. We write $\mathcal{L}_k^\ell$ for the set of all graphs $G \in \mathcal{L}_k$ satisfying $\mathrm{ml}(G - v) = \ell$ for all $v \in V(G)$, where $\ell \in \{k - 1, k\}$. (Note that $\mathcal{L}_k \neq \mathcal{L}_k^k \cup \mathcal{L}_k^{k-1}$, since there exist graphs the vertex-deleted subgraphs of which have non-constant minimum leaf number.) Leaf-guaranteed graphs offer a common framework for a series of important graph families. E.g. $\mathcal{L}_k^k$ and $\mathcal{L}_k^{k-1}$ are the *leaf-stable* and *leaf-critical* graphs defined in [9], respectively. Leaf-critical graphs are generalisations of both hypohamiltonian and hypotraceable graphs (about which see the survey of Holton and Sheehan [6, Chapter 7]). The family $\mathcal{L}_1^1$ is known as 1-*hamiltonian* graphs, a classical notion in hamiltonicity theory [2]. In applications, these graphs are often called 1-*vertex fault-tolerant* [7, Chapter 12]. $\mathcal{L}_2 \cup \mathcal{L}_3^2$ are exactly the so-called *platypus graphs* [10, 4].

In a graph G, we will call a spanning tree or hamiltonian cycle S with $\mathrm{ml}(S) = \mathrm{ml}(G)$ an *ml-subgraph*. From an application-oriented perspective it is important to point out that when a node drops from the network, the ml-subgraph used in the fault-free network may require changing equipment (we see vertices of different degrees as requiring different equipment in the network) in many nodes in order to obtain an ml-subgraph in the faulty network, which is undesirable. We formalise this by introducing, for a spanning tree or hamiltonian cycle S of G and a spanning tree or hamiltonian cycle S_v of $G - v$, the *transition cost from S to S_v* as

$$\tau(S, S_v) := |\{w \in V(G) \setminus \{v\} : \deg_S(w) \neq \deg_{S_v}(w)\}|.$$

Thus, this is the number of vertices in G which need to receive different equipment after the loss of a node; the lost node is ignored in this process. For a given graph G, denote by $\mathcal{S}_{\mathrm{ml}}(G)$ the set of all of its ml-subgraphs. In order to quantify the optimal solution in a worst-case scenario for the network itself, we introduce for a given graph G and an ml-subgraph S of G the quantity

$$\varphi_S(G) := \max_{v \in V(G)} \min_{S_v \in \mathcal{S}_{\mathrm{ml}}(G-v)} \tau(S, S_v).$$

Based on this, we shall consider the *fault cost* of the graph G representing the network:

$$\varphi(G) := \min_{S \in \mathcal{S}_{\mathrm{ml}}(G)} \varphi_S(G).$$

Graphs with fault cost 0 are easy to find: all 1-hamiltonian graphs possess this property. It is not difficult to prove that actually these are the only such graphs.

Claim 1 *A graph has fault cost 0 if and only if it is 1-hamiltonian.*

The most natural questions concerning the fault cost are therefore whether it can be arbitrarily high and whether there exist graphs with fault cost 1. While the answer for the first question is not so difficult to find, the second one is somewhat more challenging. In what follows we answer these questions affirmatively and also give a characterization of a certain subfamily of graphs with fault cost 1.

Theorem 2 *For every t there exists a graph G with $\varphi(G) > t$.*

When looking for graphs with fault cost 1, the first family to go over is graphs with minimum leaf number 2 (that is, traceable, but not hamiltonian graphs). For traceable fault cost 1 graphs (that must also be 2-leaf-guaranteed, obviously) we will use the shorthand name tfc1 graphs in the sequel.

Claim 3 *A 2-leaf-stable graph G has fault cost 1 if and only if there exist vertices $a_1, a_2 \in V(G)$, such that there exists an a_1a_2 hamiltonian path in G and for any vertex $x \in V(G) - a_1 - a_2$ there exists an a_1a_2 hamiltonian path in $G - x$ as well.*

Remark 4 *The vertices a_1 and a_2 do not have to be unique.*

An immediate corollary of the previous claim is that tfc1 graphs can have at most 2 vertices of degree 2 (by deleting a neighbour of a vertex $x \notin \{a_1, a_2\}$ of degree 2, we cannot have an a_1a_2 hamiltonian path). On the other hand, a_1 and a_2 might have degree 2 (and also any degree greater than 1), as we shall see later. This also means that tfc1 graphs need not be 3-connected. Now we are dealing with tfc1 graphs of connectivity 2, for which we need the following notions. Let X be a 2-cut of a graph G and let H be one of the components of $G - X$. Then $G[V(H) \cup X]$ is called a 2-*fragment* of G, and X is called the *attachment* of H. Let G_1 and G_2 be graphs, such that there exist two vertices x, y, such that $\{x, y\} = V(G_1) \cap V(G_2)$. Then $G_1 : G_2$ denotes the graph obtained by *gluing together* G_1 and G_2 at the vertices x, y, i.e. the graph with vertex set $V(G_1) \cup V(G_2)$ and edge set $E(G_1) \cup E(G_2)$. The next claim follows easily from Claim 3.

Claim 5 *Let G be a tfc1 graph, a_1, a_2 as described in Claim 3, and let $\{x, y\}$ be a cut of G. Then the following hold.*

1. *$\{a_1, a_2\} \cap \{x, y\} = \emptyset$*
2. *There are exactly two different 2-fragments of G with attachment $\{x, y\}$, one containing a_1 (let's call it G_1) and the other one containing a_2 (let it be G_2).*
3. *There exists an a_ix hamiltonian path in $G_i - y$ and an a_iy hamiltonian path in $G_i - x$ for $i = 1, 2$.*
4. *For any $v \in V(G_i)$ there exists an a_ix or a_iy hamiltonian path in at least one of the graphs $G_i - v$, $G_i - x - v$, $G_i - y - v$ for $i = 1, 2$.*
5. *If $(x, y) \notin E(G)$ then $G + (x, y)$ is also a tfc1 graph.*

Remark 6 *Point 3 of the claim is actually a special case of point 4, but it is worth mentioning it in its own right, because of its corollaries.*

2 Existence of tfc1 graphs

Naturally, our first aim here is to show that tfc1 graphs exist. In order to do so, let us observe that we might assume that x and y are neighbours in a 2-fragment of a tfc1 graph, by point 5 of Claim 5. If we assume this, point 4 of Claim 5 becomes much easier to handle:

Claim 7 *Let G_1, G_2, x, y, a_1, a_2 be as described in Claim 5, such that $(x, y) \in E(G)$. Then there exists an a_ix or a_iy hamiltonian path in G_i and also in $G_i - v$ for any $v \in V(G_i) - a_i$ for $i = 1, 2$.*

It seems obvious that a graph fulfilling the property described in Claim 7 is not necessarily a 2-fragment of some tfc1 graph, we need further properties. Somewhat surprisingly, these properties are easy to describe and might not even be needed (at least not in both fragments of a tfc1 graph).

Definition 8 *Let H be a connected graph, $a, x, y \in V(H)$, $(x, y) \in E(H)$. Consider the following properties of the quadruple (H, a, x, y).*

(P_0) For any $v \in V(H) - a$ there exists an ax or ay hamiltonian path in H and also in $H - v$.

(Q_1) There exists no xy hamiltonian path in H.

(Q_2) For any $v \in V(H)$ there exists no xy hamiltonian path in $H - v$.

The quadruple (H, a, x, y) is said to be a weak fragment *if it fulfills (P_0), a* medium fragment *if it fulfills (P_0) and (Q_1), and finally a* strong fragment *if it fulfills (P_0), (Q_1), and (Q_2).*

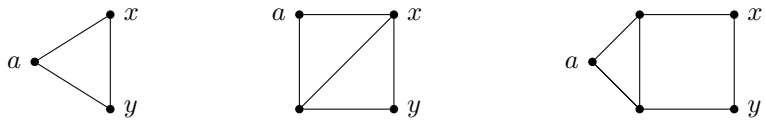


Figure 1: Examples of weak fragments.

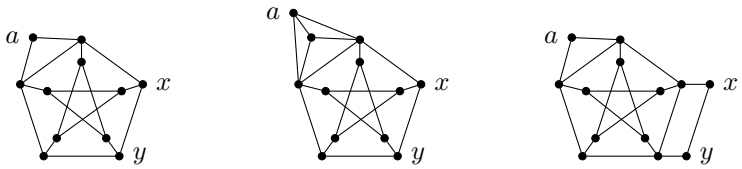


Figure 2: Examples of medium fragments.

For convenience's sake, a graph H can also be called a weak/medium/strong fragment if there exist vertices $a, x, y \in V(H)$, such that (H, a, x, y) is a weak/medium/strong fragment. By gluing together such fragments we can obtain tfc1 graphs:

Theorem 9 *Let (G_1, a_1, x, y) and (G_2, a_2, x, y) be weak fragments. If both of them are also medium or one of them is also strong, then $G_1 : G_2$ is a tfc1 graph.*

Weak fragments are easy to find, e.g. any complete graph (of order at least 3) is a weak fragment (and obviously, by adding an edge to a weak fragment we also obtain a weak fragment), see some examples in Figure 1. However, weak fragments are not enough to build tfc1 graphs using Theorem 9, we need at least one medium fragment, which is somewhat harder to find. Some examples based on Petersen's graph are shown in Figure 2. Using two (not necessarily different) graphs of Figure 2 and Theorem 9, we obtain tfc1 graphs, see Figure 3.
Next we would like to find strong fragments, which is obviously even harder than finding medium ones. First we characterize tfc1 graphs with a cut $\{x, y\}$, such that x and y are neighbours. The next theorem shows that these can only be obtained in the way described in Theorem 9.

Theorem 10 *Let $G, x, y, a_1, a_2, G_1, G_2$ be as described in Claim 3 and $(x, y) \in E(G)$. Then (G_1, a_1, x, y) and (G_2, a_2, x, y) are weak fragments and either both of them are also medium or one of them is also strong.*

Let us consider now (say) the first tfc1 graph of Figure 3 and its 2-cut X consisting of the neighbours of a_1. By Theorem 10 (which can be used, since the vertices in X are neighbours), the 2-fragments with attachment X are weak fragments, and it is obvious that K_3 (one of the fragments) is not a medium fragment, therefore the other one (which is just the first graph of the figure with a_1 deleted) must be a strong fragment.

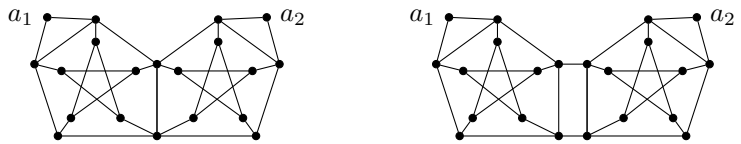


Figure 3: Examples of tfc1 graphs.

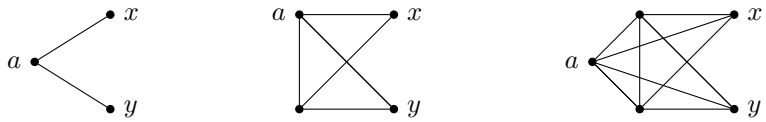


Figure 4: Weak $(1,1,1,0,0),(1,1,0,1,1),(1,1,1,1,1,)$ fragments, respectively.

3 Non-neighbouring vertices of the attachment

In the previous section we have assumed that the vertices of attachment are neighbours in a tfc1 2-fragment, in order to make the construction of tfc1 graphs easier. However, there might be tfc1 2-fragments without this property, thus the following questions arise naturally. Are there tfc1 graphs with a 2-cut consisting of non-neighbouring vertices? Are there tfc1 graphs, such that all 2-cuts consist of non-neighbouring vertices? The first question can be immediately answered in the affirmative: the second graph of Figure 3 has such a 2-cut. The corresponding (isomorphic) 2-fragments can be seen in Figure 5 (second graph). In order to also answer the second question affirmatively, we need tfc1 graphs that are not built from fragments defined in the previous section. Let us consider the following properties of a quadruple (H,a,x,y), where H is a connected, but not necessarily 2-connected graph, $a,x,y \in V(H)$.

(R_0) For any $v \in V(H)-a$ there exists an ax or ay hamiltonian path in at least one of the graphs $H-v$, $H-x-v$, $H-y-v$.

(R_1) For any $v \in V(H)-a$ there exists an ax or ay hamiltonian path in $H-v$ or an ax hamiltonian path in $H-y-v$.

(R_2) For any $v \in V(H)-a$ there exists an ax or ay hamiltonian path in $H-v$ or an ay hamiltonian path in $H-x-v$.

(R_3) For any $v \in V(H)-a$ there exists an ax or ay hamiltonian path in $H-v$.

(R_4) There exists an ax hamiltonian path in H.

(R_5) There exists an ay hamiltonian path in H.

Definition 11 *Let b_1,b_2,b_3,b_4,b_5 be integers, such that $0 \le b_1,b_2,b_3,b_4,b_5 \le 1$. A quadruple (H,a,x,y) is called a* (b_1,b_2,b_3,b_4,b_5) fragment *if it fulfills property (R_0) and for $1 \le i \le 5$, $b_i = 1$ if and only if it fulfills property (R_i).*

Again, for convenience's sake, a graph H can also be called a (b_1,b_2,b_3,b_4,b_5) fragment if there exist vertices $a,x,y \in V(H)$, such that (H,a,x,y) is a (b_1,b_2,b_3,b_4,b_5) fragment. We also define the terms medium and strong for these fragments, similarly to Definition 8 and also use the term weak (b_1,b_2,b_3,b_4,b_5) fragment for (b_1,b_2,b_3,b_4,b_5) fragments without any further requirements.

Definition 12 *A (b_1,b_2,b_3,b_4,b_5) fragment H is* medium *if it fulfills property (Q_1) (i.e. there is no xy hamiltonian path in H) and* strong *if furthermore fulfills property (Q_2) (i.e. for each $v \in V(H)$ there is no xy hamiltonian path in $H-v$).*

Obviously, (R_3) implies (R_1) and (R_2), so if $b_3 = 1$ in a (b_1,b_2,b_3,b_4,b_5) fragment, then $b_1 = b_2 = 1$. It is easy to see that weak/medium/strong fragments defined in the previous section are weak/medium/strong $(1,1,1,1,1)$ fragments, since (R_0) and $(x,y) \in V(H)$ imply (R_1), (R_2), (R_3), (R_4) and (R_5).
Weak examples, where x and y are not neighbours are easy to find: the edge deleted complete graphs of Figure 4 are $(1,1,1,0,0),(1,1,0,1,1),(1,1,1,1,1,)$ fragments, respectively. Some examples of medium $(1,1,1,0,0),(0,1,0,1,0)$, and $(1,1,1,1,1)$ fragments are shown in Figure 5. Notice that again all of them are based on Petersen's graph. This is not so apparent for the last graph, but it is obtained by deleting

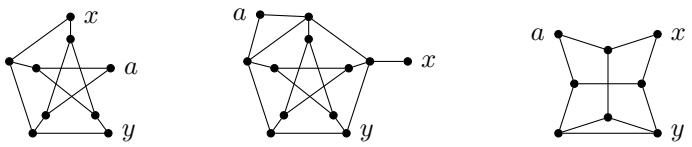


Figure 5: Medium $(1,1,1,0,0),(0,1,0,1,0),(1,1,1,1,1)$ fragments, respectively.

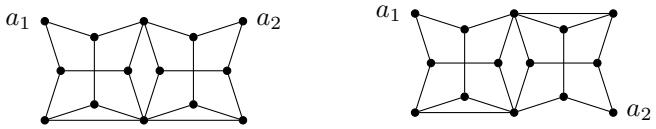


Figure 6: tfc1 graphs of order 14 without a 2-cut of neighbouring vertices.

two neighbouring vertices of Petersen's graph then adding an edge between two vertices of degree 2 having a common neighbour.
Now we are ready to generalize Theorem 9 and obtain tfc1 graphs without a 2-cut of neighbouring vertices.

Theorem 13 *Let (G_1,a_1,x,y) be a (b_1,b_2,b_3,b_4,b_5) fragment and (G_2,a_2,x,y) be a (c_1,c_2,c_3,c_4,c_5) fragment, such that both of them are medium or at least one of them is strong. If furthermore the following inequalities hold, then $G_1:G_2$ is a tfc1 graph.*

(i) $b_1+c_5\geq 1$

(ii) $b_2+c_4\geq 1$

(iii) $b_3+c_4+c_5\geq 1$

(iv) $c_1+b_5\geq 1$

(v) $c_2+b_4\geq 1$

(vi) $c_3+b_4+b_5\geq 1$

(vii) $b_4+b_5+c_4+c_5\geq 1$

Using Theorem 13 and the medium $(1,1,1,0,0)$ and/or $(1,1,1,1,1)$ fragments of Figure 5 we can create tfc1 graphs without a 2-cut consisting of neighbouring vertices. Two of these that are based on two copies of the medium $(1,1,1,1,1)$ fragment can be seen in Figure 6. These are the smallest tfc1 graphs we found (of order 14).
We also generalize Theorem 10 to obtain a characterization of all tfc1 graphs (and not just the ones with a 2-cut of neigbouring vertices).

Theorem 14 *Let G,x,y,a_1,a_2,G_1,G_2 be as described in Claim 3. Then (G_1,a_1,x,y) and (G_2,a_2,x,y) are (b_1,b_2,b_3,b_4,b_5) and (c_1,c_2,c_3,c_4,c_5) fragments for some $b_i,c_i,1\leq i\leq 5$, such that both of them are medium or at least one of them is strong and the following inequalities hold.*

(i) $b_1+c_5\geq 1$

(ii) $b_2+c_4\geq 1$

(iii) $b_3+c_4+c_5\geq 1$

(iv) $c_1 + b_5 \geq 1$

(v) $c_2 + b_4 \geq 1$

(vi) $c_3 + b_4 + b_5 \geq 1$

(vii) $b_4 + b_5 + c_4 + c_5 \geq 1$

Similarly to Theorem 10, Theorem 14 can also be used to find strong $(b_1, b_2, b_3, b_4, b_5)$ fragments. Let us consider the tfc1 graphs of Figure 6 and their 2-cuts consisting of the neighbours of a_1. By Theorem G10, the 2-fragments with attachment X are $(b_1, b_2, b_3, b_4, b_5)$ fragments (for some numbers b_i), and it is obvious that the 3 vertex path (one of the fragments) is not a medium fragment, therefore the other ones must be strong $(b_1, b_2, b_3, b_4, b_5)$ fragments. (The actual values are $(1, 1, 1, 1, 1)$ for both fragments.)

4 Open problems

In the previous sections we have found tfc1 graphs and characterized tfc1 graphs of connectivity 2. All our constructions are based on 2-fragments, thus the resulting tfc1 graphs are all of connectivity 2, therefore it is natural to ask whether 3-connected tfc1 graphs exist. If they do, is there a characterization for (say) the connectivity 3 case? Another pretty natural question is whether k-leaf-guaranteed graphs with fault cost 1 exist for $k \geq 3$.

References

[1] D. Binkele-Raible, H. Fernau, S. Gaspers, and M. Liedloff, Exact and Parameterized Algorithms for Max Internal Spanning Tree, *Algorithmica* **65** (2013) 95–128.

[2] G. Chartrand, S. F. Kapoor, and D. R. Lick, n-Hamiltonian graphs, *J. Combin. Theory, Ser. B.* **9** (1970) 308–312.

[3] L. Gargano, M. Hammar, P. Hell, L. Stacho, and U. Vaccaro, Spanning spiders and light-splitting switches, *Discrete Math.* **285** (2004) 83–95.

[4] J. Goedgebeur, A. Neyt, and C. T. Zamfirescu, Structural and computational results on platypus graphs, *Appl. Math. Comput.* **386** (2020) Article 125491.

[5] J. Goedgebeur, K. Ozeki, N. Van Cleemput, and G. Wiener, On the minimum leaf number of cubic graphs, *Discrete Math.* **342** (2019) 3000–3005.

[6] D. A. Holton and J. Sheehan, The Petersen Graph, Cambridge University Press, NY, USA (1993).

[7] L.-H. Hsu and C.-K. Lin, Graph Theory and Interconnection Networks, CRC Press, Boca Raton, FL, USA (2009).

[8] G. Salamon and G. Wiener, On finding spanning trees with few leaves, *Inform. Proc. Lett.* **105** (2008) 164–169.

[9] G. Wiener, Leaf-Critical and Leaf-Stable Graphs, *J. Graph Theory* **84** (2017) 443–459.

[10] C. T. Zamfirescu, On Non-Hamiltonian Graphs for which every Vertex-Deleted Subgraph Is Traceable, *J. Graph Theory* **86** (2017) 223–243.

On the Number of Maximal Cliques in Two-Dimensional Random Geometric Graphs: Euclidean and Hyperbolic

Hodaka Yamaji

Graduate School of Information Science and Technology
The University of Tokyo
Tokyo 113-8656, Japan
hodakaymj@g.ecc.u-tokyo.ac.jp

Abstract: Maximal clique enumeration appears in various real-world networks, such as social networks and protein-protein interaction networks for different applications. For general graph inputs, the number of maximal cliques can be up to $3^{|V|/3}$. However, many previous works suggest that the number is much smaller than that on real-world networks, and polynomial-delay algorithms enable us to enumerate them in a realistic-time span. To bridge the gap between the worst case and practice, we consider the number of maximal cliques in two popular models of real-world networks: Euclidean random geometric graphs and hyperbolic random graphs. We show that the number of maximal cliques on Euclidean random geometric graphs is lower and upper bounded by $\exp(O(|V|^{1/3}))$ and $\exp(\Omega(|V|^{1/3+\epsilon}))$ with high probability for any $\epsilon > 0$. For a hyperbolic random graph, we give the bounds of $\exp(O(|V|^{(3-\gamma)/2}))$ and $\exp(\Omega(|V|^{(3-\gamma+\epsilon)/6}))$ where γ is the power-law degree exponent between 2 and 3.

Keywords: Maximal Cliques, Random Geometric Graphs, Real-World Networks

1 Introduction

Detecting all maximal cliques in a graph is a crucial analysis tool for real-world networks from various fields: social networks, protein-protein interaction networks, and web graphs because cliques correspond to meaningful components in the networks [22, 21, 3]. Not only does it have many direct applications, but its algorithms and techniques are used in other clique-related methods such as clique percolation [19] and k-clique counting [17]. This is because we can detect all cliques by enumerating only the maximal ones.

For general graph inputs, the number of maximal cliques $\mathcal{M}$ can be up to $3^{|V|/3}$ [20]. Therefore, enumerating all of them is NP-hard. However, many studies report that in real-world networks, $\mathcal{M}$ is much smaller than that. Thus, polynomial-delay algorithms, the running time of which is bounded by $poly(|V|) \cdot \mathcal{M}$, are able to enumerate all maximal cliques in realistic-time span even for networks with millions of vertices [8]. Also, classic Bron-Kerbosch algorithm [5] (plus graph orientation [7]) is known to be efficient in many instances [9], although its worst running time is $O^*(3^{|V|/3})$ and not bounded in terms of $\mathcal{M}$. Here we strike upon the question: **why is the number of maximal cliques small on real-world networks?**

In the study of real-world networks, networks that appear naturally in various fields are considered. In terms of the graph structure, it seems that networks from different domains are completely different from each other. However, it is known that they share certain common properties. For example, they have a power-law degree distribution: the number of nodes with a vertex degree of k is proportional to $k^{-\gamma}$. In many cases, γ is between two and three, and these networks are called scale-free. Additionally,

they have the triadic closure property, meaning that if two vertices have common neighbors, they are likely to be connected. The property is often described with a measure called the clustering coefficient, and real-world graphs often have a high clustering coefficient. Other common properties include tree-like structures, small diameter, and small clique number.

One of the most combinatorially studied models of real-world networks is hyperbolic random graphs [18]. The graph is generated by independently placing vertices according to a particular distribution in a two-dimensional space with negative curvature and connecting two vertices within a certain distance. It is not so easy to construct a random graph model which satisfies both power-law degree distribution and high clustering coefficient with high probability. For example, famous models such as Erdős-Rényi [10], Wattz-Strogatz [24], Barabási-Albert [2], and Chung-Lu random graphs [1] do not have both properties. However, hyperbolic random graphs achieve that with their simple generation process. Parameters studied in this model include: the number of k-cliques, clique number [13], treewidth [4], modularity [6], and diameter [14].

In this paper, we consider the number of maximal cliques in hyperbolic random graphs. We also consider the number on two-dimensional Euclidean random geometric graphs, which are the Euclidean counterpart to hyperbolic random graphs. Euclidean random geometric graphs are also thought to be good representations of some types of real-world graphs [15, 16], although they do not possess power-law degree distribution. Our findings are as follows:

Theorem 1 (Main 1) *Let $r < 1$ be a constant. Let $\mathcal{M}$ be the number of maximal cliques in a two-dimensional Euclidean random geometric graph whose connection distance is r. There exists positive constants C_1 and C_2 such that for all $\epsilon > 0$,*

$$\Pr[\exp(C_1|V|^{1/3}) \leq \mathcal{M} \leq \exp(C_2|V|^{1/3+\epsilon})] \to 1$$

as $|V| \to \infty$.

Theorem 2 (Main 2) *Let $\gamma \in (2,3)$. Let $\mathcal{M}$ be the number of maximal cliques in a hyperbolic random graph whose power-law degree exponent is γ. There exist positive constants C_1 and C_2 such that for all $\epsilon > 0$,*

$$\Pr[\exp(C_1|V|^{(3-\gamma)/6}) \leq \mathcal{M} \leq \exp(C_2|V|^{(3-\gamma)/6+\epsilon})] \to 1$$

as $|V| \to \infty$.

For hyperbolic random graphs, we consider the case when $\gamma \in (2,3)$. In this case, the graphs are scale-free and have $\exp(\Omega(|V|^{(3-\gamma)/2}))$ cliques with high probability [13]. In general graphs, the number of maximal cliques can be up to $3^{|V|/3}$, and the bound we obtained is much smaller than that.

To prove the main theorem, we consider what is called an octahedral graph O_t. The definition of the graph is the following. Let $tK_2 = (V, E)$ where $V = \{1, 2, ..., 2t\}$ and $E = \{(i, i+t) : 1 \leq i \leq t\}$. Therefore, tK_2 is a graph with t pairwise disjoint edges. An octahedral graph O_t is the complement of tK_2. We have the following theorems from the previous study.

Theorem 3 (Forklore) *If the graph has O_t as a vertex-induced subgraph, then the number of maximal cliques is lower-bounded by 2^t.*

Theorem 4 (M. Farber, M. Hujter, and Z. Tuza [11]) *If $|V| \geq 4t$ and there exists no O_{t+1} as a vertex-induced subgraph, then the number of maximal cliques is upper-bounded by $(|V|/t)^{2t}$.*

Let $\tau(G)$ be the maximum t such that a graph G has O_t as its vertex-induced subgraph. With these theorems, all that remains is to bound τ. If τ is a constant, then the number of maximal cliques is polynomial. Unfortunately, this is not the case for the two random geometric graphs. However, our bounds on τ are much smaller than the obvious $O(|V|)$ bound.

Intuitively, τ is small because any pair of unconnected vertices have $2t-2$ common neighbors, which is against the triadic closure property i.e. two vertices with common neighbors are likely to be connected. As t gets larger, the more severe the violation becomes. This explanation can be mathematically justified on Euclidean and hyperbolic random geometric graphs. The arguments on the two different random graphs are basically the same even though the definitions of distance are different, and the parallel postulate does not hold on a hyperbolic plane.

Our contributions are as follows. Firstly, to the best of our knowledge, this is the first work that assesses the number of maximal cliques on random geometric graphs. We shed light on the importance of O_t and develop geometric and probabilistic techniques to determine its size. Those techniques are applicable to both Euclidean and hyperbolic planes. Secondly, what we have found is yet another result followed by c-closed graphs [12] supporting that the triadic closure property plays an essential role in maximal cliques. Lastly, we give an upper bound of τ on real-world graph datasets. It turns out that τ is often at most 5-20, even on networks with hundreds of thousands of vertices (See Table 1 at the end of this paper). The upper bound of $\mathcal{M}$ given by τ is still far away from the actual value. However, it is still surprising how small τ is in practice.

We are interested in whether the property of O_t has positive effects on clique enumeration algorithms on real-world graphs. Also, the generalization to a higher dimensional space is an open question.

The rest of this paper is organized as follows. In Section 2, we define Euclidean random geometric graphs and prove the main theorem. In Section 3, we define hyperbolic random graphs and discuss how the proof of the random geometric graphs on a Euclidean plane can be extended to a hyperbolic plane.

2 Euclidean Random Geometric Graphs

Let $n \in \mathbb{N}^+$, and $r \in (0,1)$. A two-dimensional Euclidean random geometric graph $G_{n,r}$ is obtained as below:

- The vertex set is $V = \{1, 2, ..., n\}$.
- The vertices are identically and independently distributed on $[0,1]^2$ according to a probability density function $f(x,y) = 1$.
- The edge set E is given by $\{(u,v) : dist(u,v) \leq r\}$

Here, $dist(u,v) = \sqrt{(u_x - v_x)^2 + (u_y - v_y)^2}$ where (u_x, u_y) and (v_x, v_y) are the xy-coordinates of u and v respectively. From here, we often identify a vertex with its position.

Given a region U on $[0,1]^2$ (where we can perform integration), define $F(U) := \int_U f(x,y)dxdy$. $F(U)$ is equal to the probability that a vertex lies on U.

2.1 Lower Bound of the Number of Maximal Cliques

Let $k \geq 4$ be an integer. Let $o' = (1/2, 1/2)$. Consider taking a polar coordinate system whose origin is o'. Let $\theta_0 := \pi/(3k)$. For $1 \leq i \leq 2k$, Define $U_i := \{(r,\phi) : r_1 \leq r \leq r_2,\ 3i\theta_0 \leq \phi \leq (3i+1)\theta_0\}$ where $r_1 := r/\sqrt{2 + 2\cos(\theta_0/2)}$ and $r_2 := r/\sqrt{2 + 2\cos(2\theta_0)}$. With some calculations, we can confirm the following.

Proposition 5 *Let $1 \leq i \leq k$ and $1 \leq j \leq 2k$. For a vertex v on U_i and a vertex w on U_j,*

$$dist(v,w) > r \ (j = i+k)$$
$$dist(v,w) \leq r \ (j \neq i+k)$$

PROOF: Let r_v and r_w be radial coordinates of v and w, respectively. If $j = i + k$, then

$$\begin{aligned} dist(v, w) &= r_v^2 + r_w^2 - 2r_v r_w \cos \angle wo'v \\ &\geq r_v^2 + r_w^2 - 2r_v r_w \cos(\pi - \theta_0) \\ &\geq 2r_1^2 + 2r_1^2 \cos\theta_0 \\ &> r \end{aligned}$$

Otherwise, $dist(v, w) \leq 2r_2^2 + 2r_2^2 \cos 2\theta_0 \leq r$ □

Let t be the number of indices $1 \leq i \leq k$ such that both U_i and U_{i+k} have at least one vertex on themselves. Then, there exists O_t as a vertex-induced subgraph. We are left to lower bound t.

Proposition 6 *$F(U_1) = \Omega(\theta_0^3)$ as $\theta_0 \to 0$*

This is not so hard to prove since $F(U_1) = \frac{1}{2}r_2^2\theta_0 - \frac{1}{2}r_1^2\theta_0$. By applying the Taylor expansion, we obtain the proposition. The next proposition states that $\mathbb{E}[t] = \Omega(n^{1/3})$ when k is taken properly.

Proposition 7 *With sufficiently large n, there exists a positive constant c such that if $k = cn^{1/3}$, then $\mathbb{E}[t] \geq \frac{c}{2}n^{1/3}$*

PROOF: Using Proposition 6, we can confirm that if n is sufficiently large, then we have $F(U_1) \geq C/(nc^3)$ for some positive constant C. Let $p = \Pr[\text{Both } U_1 \text{ and } U_{t+1} \text{ have at least one vertex on themselves}]$. Clearly, $\mathbb{E}[t] = pk$. p can be lower bounded as $p \geq 1 - 2\Pr[U_1 \text{ has no vertex}]$. Also, $\Pr[U_1 \text{ has no vertex}] = (1 - F(U_1))^n \leq e^{-nF(U_1)} \leq e^{-C/c^3}$. Therefore, by setting c small enough, we can achieve $p \geq 1/2$. □

By combining the proposition with the Chernoff bound and Theorem 3, we obtain the main theorem regarding $\mathcal{M}(G_{n,r})$.

Corollary 8 *There exists a positive constant C such that $\Pr[\exp(Cn^{1/3}) \leq \mathcal{M}(G_{n,r})] \to 1$ as $n \to \infty$.*

2.2 Upper Bound of the Number of Maximal Cliques

Let $\overline{vw}$ denote the segment between vertices v and w on a Euclidean plane. Let $\mathcal{S} := \{\overline{vw} : v, w \in G(V), dist(v, w) > r\}$. Note that $\mathcal{S}$ is like a complement of the random graph. Two segments $\overline{v_1v_2}$ and $\overline{w_1w_2}$ in $\mathcal{S}$ are called independent if $\overline{v_1w_1}$, $\overline{v_1w_2}$, $\overline{v_2w_1}$, and $\overline{v_2w_2}$ are not in $\mathcal{S}$. Two or more segments in $\mathcal{S}$ are called independent if any pair of the segments are independent. From the definition, it is obvious that

$$\begin{gathered} \text{There is no } O_{t+1} \text{ as a vertex-induced subgraph} \\ \Leftrightarrow \text{There is no set of independent segments } S \subseteq \mathcal{S} \text{ whose cardinality is } t+1. \end{gathered}$$

Therefore, we are left to bound such t on $G_{n,r}$. With elementary geometry, we can confirm the following.

Proposition 9 *Two independent segments intersect.*

Therefore, we can define an angle between two independent segments.

Let $s := \overline{v_1v_2}$ and $s' := \overline{w_1w_2}$ be independent segments. Suppose that the counter-clockwise order of four endpoints is $v_1w_1w_2v_2$. Let q be the intersection of two segments. Define a directed angle $\angle(s, s') := \angle w_1qv_1$. For convinience, define $\angle(s, s) := 0$. It holds that $\angle(s, s') = \pi - \angle(s', s)$. Therefore, the order of the two segments matters.

From here, we consider conditions that two independent segments must satisfy when the directed angle between them is known. Again, $s = \overline{v_1v_2}, s' = \overline{w_1w_2} \in \mathcal{S}$ be independent segments. Suppose $v_1w_1v_2w_2$ is the counter-clockwise order of four endpoints. Let m be the midpoint of s. Consider taking a polar coordinate system whose origin and polar axis are m and $\overrightarrow{mv_1}$. Let (r_0, ϕ) be the polar coordinate of w_1. Note that $r_0 = dist(m, w_1)$ and $\phi = \angle w_1mv_1$. The next proposition states that if the directed angle $\angle(s, s')$ is small, then r_0 is bounded tight.

Proposition 10 *Let $0 \leq \theta_0 \leq \pi/3$. If $\angle(s, s') \leq \theta_0$, then $r_0 \in [r_1, r_2]$ where*

$$r_1 := (-1/2 + \cos\theta_0)\sqrt{\cos\theta_0}$$
$$r_2 := 3/2 - \cos\theta_0$$

PROOF: Let $\theta := \angle(s, s')$. Let q be the intersection of s and s'. Let $a := dist(v_1, v_2), b := dist(w_1, w_2), c = dist(v_1, q), d = dist(w_1, q)$. Note that a is the length of s and $a > r$ holds. The same thing stands for b. From the definition of independence of segments,

$$dist(v_1, w_2) \leq r \Rightarrow c^2 + (b-d)^2 - 2c(b-d)\cos(\pi - \theta) \leq r^2$$
$$dist(v_2, w_1) \leq r \Rightarrow (a-c)^2 + d^2 - 2(a-c)d\cos(\pi - \theta) \leq r^2$$

must hold. For the first inequality, we get

$$\begin{aligned}
&c^2 + (b-d)^2 - 2c(b-d)\cos(\pi - \theta) \leq r^2 \\
&\Leftrightarrow (c + (b-d)\cos(\theta))^2 + ((b-d)\sin(\theta))^2 \leq r^2 \\
&\Rightarrow c + (b-d)\cos(\theta) \leq r \\
&\Rightarrow c - d \leq r - b\cos(\theta)
\end{aligned}$$

With the same argument, the second inequality yields $-(r - a\cos(\theta)) \leq c - d$. Regardless of how v_1, m, q, v_2 are lined up on s, we have $r_0^2 = (c - a/2)^2 + d^2 - 2(c - a/2)d\cos(\theta)$. We can upper and lower bound r_0 as below.

$$\begin{aligned}
r_0^2 &= (a/2 - c)^2 + d^2 + 2(a/2 - c)d\cos(\theta) \\
&\leq (a/2 - c + d)^2 \\
&\leq (a/2 + r - a\cos(\theta))^2 \\
&\leq r^2(3/2 - \cos(\theta_0))^2
\end{aligned}$$

$$\begin{aligned}
r_0^2 &= (a/2 - c)^2 + d^2 + 2(a/2 - c)d\cos(\theta) \\
&\geq (a/2 - c + d)^2\cos(\theta) \\
&\geq (a/2 - r + b\cos(\theta))^2\cos(\theta) \\
&\geq r^2(-1/2 + \cos(\theta_0))^2\cos(\theta_0)
\end{aligned}$$

□

If v_1, q, m, v_2 are lined up in this order, we can bound ϕ as $0 \leq \phi \leq \theta \leq \theta_0$. By considering the other case (v_1, m, q, v_2 is lined up in this order) in the same way, we have the following.

Corollary 11 *Let $0 \leq \theta_0 \leq \pi/3$. If $\angle(s, s') \leq \theta_0$, then either w_1 or w_2 must lie on a region U_1 or a region U_2 where*

$$U_1 := \{(r_0, \phi) : r_1 \leq r_0 \leq r_2, 0 \leq \phi \leq \theta_0\}$$
$$U_2 := \{(r_0, \phi) : r_1 \leq r_0 \leq r_2, \pi \leq \phi \leq \pi + \theta_0\}$$

Then we obtain the first important lemma in our work: if the directed angle is bounded small, then the expected number of independent segments is also small.

Lemma 12 *Let s be a segment. For a constant $0 \leq \theta_0 \leq \pi/3$, let X be a number of segments s' which is independent from s and $\angle(s, s') \leq \theta_0$. Then $\mathbb{E}[X] = nO(\theta_0^3)$ as $\theta_0 \to 0$.*

PROOF: By the Corollary 11, X is at most the number of vertices in U. Therefore, $\mathbb{E}[X] \leq nF(U_0 \cup U_2)$. We are left to prove $F(U_1) = F(U_2) = O(\theta_0^3)$ as $\theta_0 \to 0$, and this can be proved using the Taylor expansion. $\square$

To make full use of the Lemma 12, we prove another lemma, which states that if there exists large O_t, then we can always take a set of segments so that its size is unneglectable, and the directed angles between them are small.

Lemma 13 *Let k be a positive integer. If there exists a set of t independent segments $S \subseteq \mathcal{S}$, then there exists a segment s' and a set of segments $S' \subseteq \mathcal{S}$ such that*

$$\begin{cases} s' \in S' \\ |S'| \geq \lceil t/k \rceil \\ \forall s'' \in S'. \angle(s', s'') \leq \pi/k \end{cases} \tag{1}$$

PROOF: For an integer $1 \leq i \leq k$, define $S_i := \left\{ s \in S : \frac{i}{k}\pi \leq \angle(s_0, s) \leq \frac{i+1}{k}\pi \right\}$. Since $\bigcup_{1 \leq i \leq k} S_i = S$, we have $\sum_{i=1}^{k} |S_i| = t$. By the pigeonhole principle, there exists an index i such that $|S_i| \geq \lceil t/k \rceil$. Consider taking the segment s' in S_i so that $\angle(s_0, s') = \min_{s'' \in S_i} \angle(s_0, s'')$. If we can prove $\angle(s', s'') \leq \angle(s'', s_0) - \angle(s', s_0)$, we are done. Let a be the intersection of s_0 and s', b be that of s_0 and s'', and c be that of s' and s''. If $a = b$, then it obviously holds. We have two other cases: v_1, a, b, v_2 are lined up on s_0 in this order, and v_1, b, a, v_2 are lined up on s_0 in this order. For the former case, $\angle a = \angle(s_0, s')$, $\angle b = \pi - \angle(s_0, s'')$ and $\angle c = \angle(s', s'')$ where $\angle a$, $\angle b$, $\angle c$ are the inner angle of $\triangle abc$. Therefore,

$$\begin{aligned} &\angle a + \angle b + \angle c = \pi \qquad (*) \\ \Leftrightarrow\ &\angle(s', s'') = \pi - (\angle(s_0, s')) - (\pi - \angle(s_0, s'')) = \angle(s_0, s'') - \angle(s_0, s') \end{aligned}$$

Here, (*) follows from the fact that the sum of the inner angles of a triangle is equal to π.
The latter case can be shown in a similar way. $\square$

We finally combine the two lemmas to upper bound τ. After that, we apply Theorem 4 to get the upper bound of the number of maximal cliques.

Theorem 14 *Let ϵ be a positive constant. Let $t = n^{1/3+\epsilon}$. Then,*

$$\Pr[G_{n,r} \textit{ has } O_t \textit{ as its vertex-induced subgraph}] \to 0$$

as $n \to \infty$.

PROOF: Let $k := n^{1/3}$ and $\theta_0 := \pi/k$. Suppose $G_{n,r}$ has O_t as its vertex-induced subgraph. Apply Lemma 13 for t and k. Then there exists a segment s' and a set of independent segments S' such that (1) holds. We claim that for a certain segment s', the probability that there exists a set of segments S' such that (1) holds is at most $\exp(-\Omega(n^\epsilon))$. By Lemma 12, the expected number of segments which is possibly in S' is bounded by $O(n\theta_0^3) = O(1)$. However, to satisfy (1), The cardinarity of S' needs be at least n^ϵ. We can apply the Chernoff bound to get the probability bound $\exp(-\Omega(n^\epsilon))$.

Finally, we apply a union bound over all segments of which there are at most $n(n-1)/2$. We have

$$\Pr[G_{n,r} \text{ has } O_t \text{ as its vertex-induced subgraph}] \leq \frac{n(n-1)}{2} \exp(-\Omega(n^\epsilon))$$

This goes to 0 as $n \to \infty$. $\square$

Corollary 15 *There exist a positive constant C such that for all $\epsilon > 0$, $\Pr[\mathcal{M}(G_{n,r}) \leq \exp(Cn^{1/3+\epsilon})] \to 1$ as $n \to \infty$*

3 Hyperbolic Random Graphs

Given $n \in \mathbb{N}^+$, $\gamma \in (2, \infty)$, and $C \in \mathbb{R}$, let $R := 2\log n + C$, and $\alpha = (\gamma - 1)/2$. A hyperbolic random graph $G_{n,\gamma,C}$ is obtained as below:

- The vertex set is $V = \{1, 2, ..., n\}$.
- The vertices are identically and independently distributed on a hyperbolic plane. The probability density function by a polar coordinate is:

$$f(r, \theta) = \begin{cases} \frac{1}{2\pi} \cdot \frac{\alpha \sinh(\alpha r)}{\cosh(\alpha R) - 1} & (r \leq R) \\ 0 & (r > R) \end{cases}$$

- The edge set E is given by $\{(u, v) : dist(u, v) \leq R\}$

In our work, we are interested in the case $2 < \gamma < 3$ $(1/2 < \alpha < 1)$ where the generated graph is "scale-free" with high probability. Let (r_u, ϕ_u) and (r_v, ϕ_v) be polar coordinates of u and v respectively. Let $\theta := \pi - |\pi - |\phi_u - \phi_v||$. Then, the hyperbolic cosine formula suggests that

$$\cosh(dist(u, v)) = \cosh r_u \cosh r_v - \sinh r_u \sinh r_v \cos\theta$$

Again, define $F(U) := \int_U f(r, \theta) dr d\theta$. On a hyperbolic plane, the area of U is equal to $\int \sinh r dr d\theta =: \mu(U)$. Let $\rho(r, \theta) := \frac{dF}{d\mu} = \frac{f(r,\theta)}{\sinh r}$. Intuitively, ρ is the the probability that a vertex lies on a single unit square around (r, θ). By differentiating ρ, we can confirm that it is monotonically decreasing with respect to r. Thus, the graph is dense near the origin of the plane.

To obtain the lower bound of $\mathcal{M}(G_{n,r,C})$, we do the same thing as the Euclidean random geometric graphs i.e. We take regions so that vertices on them form O_t. We obtain the following theorem.

Theorem 16 *There exists a positive constant C' such that* $\Pr[\exp(C' n^{(1-\alpha)/3}) \leq \mathcal{M}(G_{n,r,C})] \to 1$ *as* $n \to \infty$.

Let $\overline{vw}$ denote the segment (i.e. the shortest geodesic) between vertices v and w on a hyperbolic plane. Let $\mathcal{S} := \{\overline{vw} : v, w \in G(V), dist(v, w) > R\}$. We define the independence of segments in the same way as for the Euclidean case. We can prove that two independent segments intersect. Therefore, we can also define the angle of segments on a hyperbolic plane.

To upper bound $\mathcal{M}(G_{n,\gamma,C})$, we prove the hyperbolic versions of the Lemma 12 and 13. For the proof of the hyperbolic version of the Lemma 12, we use the hyperbolic cosine formula instead of the Euclidean one. We obtain the following, which is similar to the Corollary 11.

Corollary 17 *Let $\theta_0 \geq 0$. If $\angle(s, s') \leq \theta_0$, either w_1 or w_2 must lie on U_1 or U_2 where*

$$U_1 := \{(r_0, \phi) : r_1 \leq r_0 \leq r_2, 0 \leq \phi \leq \theta_0\}$$
$$U_2 := \{(r_0, \phi) : r_1 \leq r_0 \leq r_2, \pi \leq \phi \leq \pi + \theta_0\}$$

Here, r_1 and r_2 depend on R and θ_0, which we will not give explicitly here. As a corollary, we have

Corollary 18 *Let s be a segment. Let $\theta_0 \geq 0$ Let X be a number of segments s' which is independent from s and $\angle(s, s') \leq \theta_0$, then*

$$\mathbb{E}[X] = \sup_{(r,\theta) \in U} \{\rho(r, \theta)\} O(n^2 \theta_0^3)$$

as $\theta_0 \to 0$ and $n \to \infty$

The proof of Lemma 13 on a Euclidean plane is also valid on a hyperbolic plane, except the part (*) where we used the fact that the sum of inner angles of a triangle is equal to π. On a non-Euclidean plane without the parallel postulate, Saccheri-Legendre theorem [23] can be used. The theorem claims $\angle a + \angle b + \angle c \leq \pi$, and we can replace (*) with it.

It is not so hard from here to prove the limited version of the main theorem. The next proposition states that if vertices of O_t do not fall on a region near the origin, we can bound its size.

Proposition 19 *Let $V' := \{v \in V(G_{n,\gamma,C}) : \text{the radial coordinate of } v \text{ is greater than } R/2\}$. Let $G'_{n,\gamma,C}$ be a subgraph of $G_{n,\gamma,C}$ induced by V'. Let $t = n^{(1-\alpha)/3+\epsilon}$ where ϵ is arbitrary positive constant. Then,*

$$\Pr[G'_{n,\gamma,C} \text{ has } O_t \text{ as its vertex-induced subgraph}] \to 0$$

as $n \to \infty$.

Due to the page limitation, we omit the full proof of the main theorem.

Theorem 20 *There exist a positive constant C' such that for all $\epsilon > 0$,*

$$\Pr[\mathcal{M}(G_{n,\gamma,C}) \leq \exp(C' n^{(1-\alpha)/3+\epsilon})] \to 1$$

as $n \to \infty$

References

[1] W. Aiello, F. Chung, and L. Lu. A random graph model for massive graphs. In *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing*, pages 171-180, 2000.

[2] R. Albert, and A. L. Barabási, Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47, 2002.

[3] L. Becchetti, P. Boldi, C. Castillo, and A. Gionis. Efficient semi-streaming algorithms for local triangle counting in massive graphs. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 16-24, 2008

[4] T. Bläsius, T. Friedrich, and A. Krohmer. Hyperbolic random graphs: separators and treewidth. In *Proceedings of the 24th Annual European Symposium on Algorithms*, pages 1-16, 2016

[5] C. Bron, and J. Kerbosch. Algorithm 457: finding all cliques of an undirected graph. *Communications of the ACM*, 16(9):575-577, 1973

[6] J. Chellig, N. Fountoulakis, and F. Skerman. On the diameter of hyperbolic random graphs. *Journal of Complex Networks*, 10(1):cnab051, 2022

[7] N. Chiba, and T. Nishizeki. Arboricity and subgraph listing algorithms. *SIAM Journal on Computing*, 14(1):210-223, 1985

[8] A. Conte, R. Grossi, A. Marino, and L, Versari. Sublinear-space bounded-delay enumeration for massive network analytics: Maximal cliques. In *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016)*, volume 55, 148:1-148:15, 2016

[9] D. Eppstein, M. Löffler, and D. Strash. Listing all maximal cliques in large sparse real-world graphs. *Journal of Experimental Algorithmics*, 18:1-3, 2013

[10] P. Erdős, and A. Rényi. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, 5(1):17-60, 1960.

[11] M. Farber, M. Hujter, and Z. Tuza. An upper bound on the number of cliques in a graph. *Networks*, 23(3):207-210, 1993

[12] J. Fox, T. Roughgarden, C. Seshadhri, F. Wei, and N. Wein. Finding cliques in social networks: A new distribution-free model. *SIAM journal on computing*, 49(2):448-464, 2020

[13] T. Friedrich, and A. Krohmer. Cliques in hyperbolic random graphs. In *Proceedings of the 34th IEEE Conference on Computer Communications*, pages 1544-1552, 2015

[14] T. Friedrich, and A. Krohmer. On the diameter of hyperbolic random graphs. *SIAM Journal on Discrete Mathematics*, 32(2):1314-1334, 2018

[15] M. Haenggi, J. G. Andrews, F. Baccelli, O. Dousse, and M. Franceschetti. Stochastic geometry and random graphs for the analysis and design of wireless networks *IEEE Journal on Selected Areas in Communications*, 27(7):1029-1046, 2009

[16] D. J. Higham, M. Rašajski, and N. Pržulj. Fitting a geometric graph to a protein–protein interaction network. *Bioinformatics*, 24(8):1093-1099, 2008

[17] S. Jain, and C. Seshadhri. The power of pivoting for exact clique counting. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 268-276, 2020

[18] D. Krioukov, F. Papadopoulos, M. Kitsak, A. Vahdat, and M. Boguná. Hyperbolic geometry of complex networks. *Physical Review E*, 82(3):036106, 2010

[19] J. M. Kumpula, M. Kivelä, K. Kaski, and J. Saramäki. Sequential algorithm for fast clique percolation. *Physical Review E*, 78(2):026109, 2008

[20] J. W. Moon, and L. Moser. On cliques in graphs. *Israel Journal of Mathematics*, 3(1):23-28, 1965

[21] T. Nepusz, H. Yu, and A. Paccanaro. Detecting overlapping protein complexes in protein-protein interaction networks. *Nature Methods*, 9(5):471-472, 2012

[22] G. Palla, I. Derényi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814-818, 2005

[23] P. J. Ryan. Euclidean and non-euclidean geometry: an analytic approach. *Cambridge University Press*, 1986

[24] D. J. Watts, and S, H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440-442, 1998

Graph	$\|V\|$	$\|E\|$	τ
amazon0601	403394	2443408	4
as-skitter	1696415	11095298	14
email-EuAll	265214	364481	7
soc-Epinions1	75879	405740	9
web-Google	875713	4322051	4
web-NotreDame	325729	1090108	4
web-Stanford	281903	1992636	4
wiki-Talk	2394385	4659565	12
wiki-topcats	1791489	25444207	9

Table 1: Upper bound of τ (i.e. maximum t such that a graph contains O_t as its vertex-induced subgraph) on SNAP graph datasets

Solving the Maximum Popular Matching Problem with Matroid Constraints

Gergely Csáji[1]
Eötvös Loránd University
Budapest, Hungary
csaji.gergely@student.elte.hu

Tamás Király[2]
ELKH-ELTE Egerváry Research Group
Budapest, Hungary
tamas.kiraly@ttk.elte.hu

Yu Yokoi[3]
National Institute of Informatics
Tokyo, Japan
yokoi@nii.ac.jp

We consider the problem of finding a maximum popular matching in a many-to-many matching setting with two-sided preferences and matroid constraints. This problem was proposed by Kamiyama (2020) and solved in the special case where matroids are base orderable. Utilizing a newly shown matroid exchange property, we show that the problem is tractable for arbitrary matroids. We further investigate a different notion of popularity, where the agents vote with respect to lexicographic preferences, and show that both existence and verification problems become NP-hard, even in the b-matching case.

Keywords: popular matching, stable matching, matroid kernel

1 Introduction

The notion of *popular matching* is a natural adaptation of the notion of weak Condorcet winner [5] to the marriage model of Gale and Shapley [10], where agents of a two-sided market have strict preference orders on admissible agents on the other side. It is a well-known fact (sometimes called the Condercet paradox) that a weak Condorcet winner does not always exist in the general setting. Remarkably, existence is guaranteed in the marriage model: Gärdenfors [11] showed that every stable matching is popular. In fact, stable matchings are the smallest popular matchings, so the notion of popular matching can be considered as a relaxation of stable matching, where we sacrifice pairwise stability in order to achieve larger size.

Several years after the results of Gärdenfors, popular matchings came into the focus again in the 2000s due to their interesting algorithmic properties. Huang and Kavitha [13] showed that a maximum size popular matching in the marriage model can be found in polynomial time. In contrast, recently it was shown by Gupta et al. [12] that deciding the existence of a popular matching in the roommates (i.e., non-bipartite) model is NP-complete.

Just as in the case of the stable marriage problem, the results have been extended to many-to-many matchings. The concept of Condorcet winner is not so straightforward in this setting, because

[1]Research was supported by the Lendület Programme of the Hungarian Academy of Sciences – grant number LP2021-1/2021, by the Hungarian National Research, Development and Innovation Office – NKFIH, grant number K143858.
[2]Research was supported by the Lendület Programme of the Hungarian Academy of Sciences – grant number LP2021-1/2021, by the Hungarian National Research, Development and Innovation Office – NKFIH, grant numbers TKP2020-NKA-06 and K143858.
[3]Research was supported by JST PRESTO Grant Number JPMJPR212B.

there are several different ways in which an agent can compare two matchings based on the sets of partners. Nonetheless, remarkable findings by Brandl and Kavitha [2, 3] show that popular many-to-many matchings exist under a rather restrictive definition of popularity, and furthermore, the largest such matching has maximum size even among matchings satisfying a much less restrictive notion of popularity.

Nasre and Rawat [16] introduced a many-to-many model where agents can have classifications in their preference lists, and classes can have upper quotas. Kamiyama [14] generalized the results further, extending the laminar nested classification of Nasre and Rawat to a matroid structure. He gave an algorithm that returns a popular matching, based on Fleiner's algorithm for finding a matroid kernel [8, 9], which is a generalization of the notion of stable matching to matroid intersection.

For the maximum size popular matching problem, however, Kamiyama only gave an efficient algorithm for the special case when the matroids are weakly base orderable. He left open the question whether there is a polynomial-time algorithm that works for arbitrary matroids.

In this paper, we give an affirmative answer to this question. We show that *the maximum popular matching problem with two-sided preferences and arbitrary matroid constraints can be solved in polynomial time*, by essentially the same algorithm as in [14]. The key tool in extending the proof from weakly base orderable matroids to arbitrary matroids is a new matroid exchange property that can be formulated in terms of voting between two independent sets in an ordered matroid (Theorem 3). We prove the theorem by combining matroid techniques with the duality between weighted bipartite matching and minimum cover. We also show a property similar to the one by Brandl and Kavitha mentioned above: there always exists a matching satisfying a remarkably restrictive definition of popularity that has maximum size among all matchings satisfying weaker popularity properties.

We present our results in the framework of matroid intersection, which is equivalent to Kamiyama's model, but involves only two matroids, and allows us to better describe the difference between the more restrictive and less restrictive popularity notions. It is also closer to the original matroid kernel problem defined by Fleiner [8].

We also investigate another notion of popularity, called lexicographic popularity. Here, each agent has only one vote, and the agents compare the different matchings in a lexicographic way. Lexicographic preferences have been of considerable interest recently, as they arise in many applications. Cechlárová et al. [4] studied Pareto-optimal matchings in the many-to-many matching problem with lexicographic one-sided preferences. Biró and Csáji [1] looked at strong core and Pareto optimality with two-sided lexicographic preferences. Closest to our work is the paper of Paluch [17], which studied popular and clan-popular matchings in the many-to-one matching problem with one-sided lexicographic preferences. We show that, in contrast to the previous notion of popularity, a lexicographically popular matching does not always exist and both the search and verification questions regarding lexicographic popularity are NP-hard, even in the restricted case of b-matchings with constant degrees and capacities.

The rest of the paper is structured as follows. In Section 2, we describe the matroid kernel problem, and a matroid exchange property from our previous paper [7], whose generalization is used in our proof. In Section 3, we define the various notions of voting and popularity that we consider in the popular matroid intersection problem, and we describe their relationship to the popularity notions used in the literature on many-to-many matchings. We also present our new result on matroid exchanges, which is proved in Section 4. In Section 5, we describe the algorithm for the maximum size popular matroid intersection problem and the proof of its correctness. Finally, in Section 6 we define lexicographic popularity and provide hardness results for the related search and verification problems.

2 Ordered matroids and matroid kernels

A *matroid* is a pair $(S,\mathcal{I})$ of a finite set S and a nonempty family $\mathcal{I} \subseteq 2^S$ satisfying the following two axioms: (i) $A \subseteq B \in \mathcal{I}$ implies $A \in \mathcal{I}$, and (ii) for any $A, B \in \mathcal{I}$ with $|A| < |B|$, there is $v \in B \setminus A$ with $A + v \in \mathcal{I}$. A set in $\mathcal{I}$ is called an *independent set*, and an inclusion-wise maximal one is called a *base*. By axiom (ii), all bases have the same size, which is called the *rank* of the matroid.

A *circuit* of a matroid is an inclusionwise minimal dependent set. The *fundamental circuit* of an

element $x \in S \setminus B$ for a base B is the unique circuit in $B + x$. We will use the following well-known property.

Proposition 1 (Strong circuit axiom) *If C, C' are circuits, $x \in C \setminus C'$, and $y \in C \cap C'$, then there is a circuit $C'' \subseteq C \cup C'$ such that $x \in C''$ and $y \notin C''$.*

In our proofs in Section 4, we will use the fact that matroids are closed under operations such as *direct sum*, *restriction*, *contraction*, and *truncation*. For these operations and other basics on matroids, we refer the reader to [18].

An *ordered matroid* is a triple $(S, \mathcal{I}, \succ)$ such that $(S, \mathcal{I})$ is a matroid and $\succ$ is a linear order on S. The linear order determines an optimal base in the following sense: for any weight vector $w \in \mathbb{R}^S$ which satisfies $w_x > w_y \Leftrightarrow x \succ y$, the unique maximum weight base is the same. We call this base A the *optimal base* of $(S, \mathcal{I}, \succ)$; it is characterized by the property that $u \succ_i v$ for every $u \in A$ and $v \in S \setminus A$ for which $A - u + v \in \mathcal{I}$.

The following theorem was recently shown in [7, Theorem 3]. We provide a generalization of this result in Section 3 (Theorem 3) which plays a key role in our proofs. For clarity of presentation, we use the word 'pairing' instead of 'matching' for a family of disjoint pairs of elements from two given disjoint subsets A and B. Thus, a *pairing between A and B* is a matching in the complete bipartite graph with vertex classes A and B, while a *perfect pairing* is a perfect matching in the same graph.

Theorem 2 (Csáji, Király, Yokoi [7]) *Let $M = (S, \mathcal{I}, \succ)$ be an ordered matroid of rank r. Let A be the optimal base and B be a base disjoint from A. Then, there is a perfect pairing $a_i b_i$ ($i \in [r]$) between A and B such that $a_i \succ b_i$ and $B + a_i - b_i \in \mathcal{I}$ for every $i \in [r]$.*

Let $M_1 = (S, \mathcal{I}_1, \succ_1)$ and $M_2 = (S, \mathcal{I}_2, \succ_2)$ be ordered matroids on the same ground set S, and let $I \in \mathcal{I}_1 \cap \mathcal{I}_2$ be a common independent set. We say that an element $v \in S \setminus I$ is *dominated* by I in M_i if $I + v \notin \mathcal{I}_i$ and $u \succ_i v$ for every $u \in I$ for which $I - u + v \in \mathcal{I}_i$. We call a common independent set $I \in \mathcal{I}_1 \cap \mathcal{I}_2$ an (M_1, M_2)-*kernel* if every $v \in S \setminus I$ is dominated by I in M_1 or M_2. If an element $v \in S \setminus I$ is dominated in neither M_1 nor M_2, we say that v *blocks* I.

It was shown by Fleiner [8, 9] that matroid kernels always exist and have the same size – in fact, they have the same span in both matroids. He also gave a matroidal version of the Gale–Shapley algorithm that finds an (M_1, M_2)-kernel efficiently, in $\mathcal{O}(|S|^2)$ time.

To understand the relation between our problem formulation and the formulation of Kamiyama [14], it is instructive to see the equivalence of the matroid kernel model above and the model of *stable matchings with matroid constraints*, as described below. Let $G = (V_1, V_2; E)$ be a bipartite graph, and for each $v \in V_1 \cup V_2$, let $M_v = (\delta_G(v), \mathcal{I}_v, \succ_v)$ be an ordered matroid, where $\delta_G(v)$ denotes the set of edges incident to v. An edge set $I \subseteq E$ is called a *matching* if $I \cap \delta_G(v) \in \mathcal{I}_v$ for every $v \in V_1 \cup V_2$. A matching I is *stable* if for any $e = v_1 v_2 \in E \setminus I$, either $I \cap \delta_G(v_1)$ is the optimal base of M_{v_1} restricted to $(I + e) \cap \delta_G(v_1)$, or $I \cap \delta_G(v_2)$ is the optimal base of M_{v_2} restricted to $(I + e) \cap \delta_G(v_2)$.

We show that this is actually equivalent to the matroid kernel model. To formulate stable matchings with matroid constraints as a matroid kernels problem, let M_1 be the matroid on ground set E obtained as the direct sum of the matroids M_v ($v \in V_1$), and let $\succ_1$ be obtained by arbitrarily extending the linear orders $\succ_v$ ($v \in V_1$) into a linear order on E. We define M_2 and $\succ_2$ similarly using V_2. It is easy to see that (M_1, M_2)-kernels are exactly the stable matchings. Conversely, a matroid kernel problem can be written as a stable matching problem with matroid constraints, where G consists of two vertices and $|S|$ parallel edges between them.

We will see in the next section that the correspondence between the two models is somewhat more complicated in case of popular matchings, because we have to define the voters, which corresponds to fixing an appropriate partitioning of the ground set in both matroids in the matroid intersection model.

3 Voting and popularity in matroid intersection

3.1 Voting in ordered matroids

Consider an ordered matroid $M = (S, \mathcal{I}, \succ)$. Given an ordered pair of independent sets (I, J), let N be a pairing between $I \setminus J$ and $J \setminus I$ and consider the following two conditions:

(1) $I - u + v \in \mathcal{I}$ for every $uv \in N$, where $u \in I \setminus J$ and $v \in J \setminus I$.

(2) Any element of $J \setminus I$ spanned by I is covered by N.

We say that N is a *weakly feasible pairing* for (I, J) if (1)-(2) hold. For two independent sets I and J and a weakly feasible pairing N for (I, J), we define

$$\mathsf{vote}(I, J, N) = |\{\, uv \in N : u \succ v \,\}| - |\{\, uv \in N : u \prec v \,\}| + |I| - |J|,$$

where $u \in I \setminus J$ and $v \in J \setminus I$. Considering the most adversarial weakly feasible pairing, we define

$$\mathsf{vote}^\bullet(I, J) = \min\{\, \mathsf{vote}(I, J, N) : N \text{ is a weakly feasible pairing for } (I, J) \,\}.$$

The above definition of voting is natural in our model, but it leads to a more restricted notion of popularity than that of Kamiyama [14], because the conditions on N are weaker. The reason is that when we construct the matroid M_1 from the matroids M_v ($v \in V_1$), as described at the end of Section 2, we lose the information on the individual voters, i.e., the partition of the edge set corresponding to the vertices of V_1. Hence, in order to retrieve the popularity notion in [14], we have to introduce a definition of voting that uses that extra information, which is given as a fixed partition of S.

Let $\mathcal{P} = \{U_1, \ldots, U_k\}$ be a fixed partition of S such that each U_j is a union of some connected components of the matroid M. Given an ordered pair of independent sets (I, J), consider the following three additional conditions for a pairing N between $I \setminus J$ and $J \setminus I$:

(3) Every $uv \in N$ satisfies $u, v \in U_j$ for some $j \in [k]$.

(4) For every $j \in [k]$, the number of pairs of N induced by U_j is $\min\{|U_j \cap (I \setminus J)|, |U_j \cap (J \setminus I)|\}$.

(5) Any element of $I \setminus J$ spanned by J is covered by N.

We say that N is a *feasible pairing* for (I, J) if (1)-(5) hold. Considering the most adversarial feasible pairing, we define

$$\mathsf{vote}(I, J) = \min\{\, \mathsf{vote}(I, J, N) : N \text{ is a feasible pairing for } (I, J) \,\}.$$

It turns out that the following property is crucial for proving the main results.

Theorem 3 $\mathsf{vote}(I, J) + \mathsf{vote}(J, I) \leq 0$.

We present the proof in the next section. The following is an immediate corollary.

Corollary 4 *The following sequence of inequalities holds for any pair of independent sets I and J:* $\mathsf{vote}^\bullet(I, J) \leq \mathsf{vote}(I, J) \leq -\mathsf{vote}(J, I) \leq -\mathsf{vote}^\bullet(J, I)$.

Remark 5 *Here, we make two remarks about Theorem 3.*

First, we see that the statement can be shown easily if the matroid $(S, \mathcal{I})$ is weakly base orderable. Indeed, the definition of weak base orderability implies the existence of a pairing N that is feasible for both (I, J) and (J, I), from which we obtain $\mathsf{vote}(I, J) + \mathsf{vote}(I, J) \leq \mathsf{vote}(I, J, N) + \mathsf{vote}(J, I, N) = 0$. *For a general matroid, however, such a pairing N may not exist, which makes it difficult to extend the proof arguments in previous works [14, 15] to general matroids. We use Theorem 3 to overcome this difficulty.*

Second, Theorem 3 can be regarded as a generalization of Theorem 2. Let I and J be A and B in the statement of Theorem 2, respectively, and consider the trivial partition $\mathcal{P} = \{S\}$. By the optimality of A, we must have $\mathsf{vote}(A, B) = r$, *and then Theorem 3 implies* $\mathsf{vote}(B, A) = -r$. *This shows the existence of the perfect pairing claimed in Theorem 2.*

3.2 Popularity in matroid intersection

Let $M_1 = (S, \mathcal{I}_1, \succ_1)$ and $M_2 = (S, \mathcal{I}_2, \succ_2)$ be ordered matroids and $\mathcal{P}_1 = \{U_1^1, \ldots, U_{k_1}^1\}$ and $\mathcal{P}_2 = \{U_1^2 \ldots, U_{k_2}^2\}$ be fixed partitions of S such that each U_j^i is a union of connected components of M_i. For an ordered pair (I, J) of common independent sets and $i \in \{1, 2\}$, we define $\mathsf{vote}_i(I, J)$ as $\mathsf{vote}(I, J)$ with respect to M_i and $\mathcal{P}_i$. We call a common independent set $I \in \mathcal{I}_1 \cap \mathcal{I}_2$ *popular* if $\mathsf{vote}_1(I, J) + \mathsf{vote}_2(I, J) \geq 0$ for every $J \in \mathcal{I}_1 \cap \mathcal{I}_2$. Also, we call $I \in \mathcal{I}_1 \cap \mathcal{I}_2$ *defendable* if $\mathsf{vote}_1(J, I) + \mathsf{vote}_2(J, I) \leq 0$ for every $J \in \mathcal{I}_1 \cap \mathcal{I}_2$.

Remark 6 *It is important to remember that feasible pairings for (I, J) are not the same as feasible pairings for (J, I). When considering popularity of I, we compare it to J by taking a feasible pairing for (I, J) that is worst possible for I. In contrast, defendability of I is determined by considering a feasible pairing for (J, I) that is best possible for I.*

By using $\mathsf{vote}_i^\bullet$ instead of vote_i, we can define a stronger version of popularity and a weaker version of defendability, which we call *super popularity* and *weak defendability*, respectively. Note that these do not depend on the partitions $\mathcal{P}_1$ and $\mathcal{P}_2$. The relation between these notions can be derived from Theorem 3.

Corollary 7 *The following implications hold for any $I \in \mathcal{I}_1 \cap \mathcal{I}_2$:*

$$I \text{ is super popular} \Rightarrow I \text{ is popular} \Rightarrow I \text{ is defendable} \Rightarrow I \text{ is weakly defendable}$$

Proof: It follows from Corollary 4 that

$$\begin{aligned}\mathsf{vote}_1^\bullet(I, J) + \mathsf{vote}_2^\bullet(I, J) &\leq \mathsf{vote}_1(I, J) + \mathsf{vote}_2(I, J) \\ &\leq -\mathsf{vote}_1(J, I) - \mathsf{vote}_2(J, I) \leq -\mathsf{vote}_1^\bullet(J, I) - \mathsf{vote}_2^\bullet(J, I)\end{aligned}$$

for any $J \in \mathcal{I}_1 \cap \mathcal{I}_2$. This gives the required implications. □

In Section 4, we prove Theorem 3. In Section 5, we show that an abstract version of Kamiyama's algorithm [14] outputs a common independent set that is super popular, and largest among all weakly defendable common independent sets. This generalizes several results in previous works. In Kamiyama's model [14], feasible pairings are defined by conditions (1)-(4). Then, our result shows that the algorithm's output is a largest popular common independent set also in his definition. In the partition matroid case (i.e., b-matching case) studied by Brandl-Kavitha [2], our popularity notion coincides with their popularity and our defendability coincides with their weak popularity. Therefore, our result generalizes the result of Brandl-Kavitha [2] that we can efficiently find a popular matching that is largest among all weakly popular matchings.

4 Proof of Theorem 3

Recall that $M = (S, \mathcal{I}, \succ)$ is an ordered matroid, and $\mathcal{P} = \{U_1, \ldots, U_k\}$ is a fixed partition of S such that each U_j is a union of some connected components of the matroid M. Let $I \in \mathcal{I}$ and $J \in \mathcal{I}$ be arbitrary independent sets. Our aim is to prove that $\mathsf{vote}(I, J) + \mathsf{vote}(J, I) \leq 0$.

For a member U_j of the partition $\mathcal{P}$, let $I_j := U_j \cap I$ and $J_j := U_j \cap J$. If $|I_j| \leq |J_j|$, then let $A_j \subseteq J_j \setminus I_j$ be a set satisfying $I_j \cup A_j \in \mathcal{I}$ and $|A_j| = |J_j| - |I_j|$, and set $I'_j := I_j \setminus J_j$ and $J'_j := J_j \setminus (I_j \cup A_j)$. If $|I_j| > |J_j|$, then define I'_j and J'_j similarly by exchanging the roles of I_j and J_j. In any case, we have $|I'_j| = |J'_j| = \min\{|U_j \cap (I \setminus J)|, |U_j \cap (J \setminus I)|\}$. Let M_j be the matroid obtained by restricting M to $I_j \cup J_j$, contracting $(I_j \cap J_j) \cup A_j$, and truncating to the size of $|I'_j|$. The ground set of M_j is partitioned into two bases I'_j and J'_j. Let $M' = (S', \mathcal{I}')$ be the direct sum of $M_1, \ldots, M_k$. The ground set S' of M' is partitioned into two bases $I' := I'_1 \cup \cdots \cup I'_k$ and $J' := J'_1 \cup \cdots \cup J'_k$.

Let $G_I = (I', J'; E_I)$ be the bipartite graph with $E_I = \{\, uv : u \in I', v \in J', I' - u + v \in \mathcal{I}' \,\}$, and let $G_J = (I', J'; E_J)$ where $E_J = \{\, uv : u \in I', v \in J', J' + u - v \in \mathcal{I}' \,\}$. Since I' and J' are bases of M', both G_I and G_J are perfectly matchable.

Claim 8 *Any perfect matching of G_I is a feasible pairing for (I, J), and any perfect matching of G_J is a feasible pairing for (J, I).*

PROOF: By symmetry, it is enough to prove the first statement. Let N be a perfect matching in G_I. By definition, N is a pairing between $I \setminus J$ and $J \setminus I$. We show that N satisfies conditions (1)-(5).

To see (1), consider $uv \in N$ such that $u \in I'_j$ and $v \in J'_j$. Then $uv \in E_I$ implies that $I'_j - u + v$ is independent in M_j, so $(I \cap U_j) - u + v$ is independent in M by the construction of M_j. Since U_j is the union of some components of M, this implies $I - u + v \in \mathcal{I}$.

To show (2), consider an element $v \in J \setminus I$ not covered by N. Then $v \in A_j$ for some j such that $|J_j| > |I_j|$. By definition, $A_j \cup I_j$ is independent in M, so I_j cannot span v. Since U_j is the union of connected components, I cannot span v either.

Conditions (3) and (4) are satisfied by definition, and the proof of (5) is similar to (2), by exchanging the role of I and J. □

For $uv \in E_I$, let $w(uv) = 1$ if $u \prec v$, and $w(uv) = 0$ if $u \succ v$, and let k be the maximum weight of a perfect matching in E_I. Then $\mathsf{vote}(I, J) \leq |I'| - 2k$. By duality, there is an integer function π on S' such that $\sum_{v \in S'} \pi(v) = k$ and $\pi(u) + \pi(v) \geq w(uv)$ for every $uv \in E_I$.

We now consider the same weight function on E_J: let $w(uv) = 1$ if $u \prec v$, and $w(uv) = 0$ if $u \succ v$. Let E consist of the edges $uv \in E_J$ which satisfy $\pi(u) + \pi(v) \geq w(uv)$.

Lemma 9 *The bipartite graph $G = (I', J'; E)$ has a perfect matching.*

PROOF: In the proof, we work with the matroid M', so the term 'circuit' refers to circuits of M'. Suppose for contradiction that there exists a subset X of I' such that the set of its neighbors in G, that we denote by Y, is smaller than X. We introduce a new ordering $\succ'$ on the elements of S': $a \succ' b$ if either $\pi(a) < \pi(b)$, or $\pi(a) = \pi(b)$ and $a \succ b$ (we will only compare pairs inside I' or inside J'). The following claim is the main ingredient of the proof.

Claim 10 *Let C be a circuit of M' such that $C \cap I' \subseteq X$, and let v be the worst element of $C \cap J'$ according to $\succ'$. Then $v \in Y$.*

PROOF: Suppose for contradiction that $v \notin Y$. There must exist a vertex $u \in C \cap X$ such that $uv \in E_J$. Indeed, otherwise we could eliminate the elements of $C \cap X$ one by one using the strong circuit axiom with the fundamental circuits for J', while retaining the property that v is in the circuit; in the end, we would obtain a circuit inside J', which is impossible. So, there is a vertex $u \in C \cap X$ such that $uv \in E_J$. Let us call a vertex $v' \in J'$ *bad* if either $\pi(u) + \pi(v') < 0$, or $\pi(u) + \pi(v') = 0$ and $u \prec v'$. The vertex v is bad because $uv \in E_J$ and $v \notin Y$. Since v is the worst element of $C \cap J'$ according to $\succ'$, we have that every $v' \in C \cap J'$ is bad. Thus, $uv' \notin E_I$ holds for every $v' \in C \cap J'$ (since $\pi(u) + \pi(v') \geq w(uv')$ if $uv' \in E_I$). In other words, u is not in the fundamental circuit of v' for I' for any $v' \in C \cap J'$. But then we could eliminate the elements of $C \cap J'$ one by one using the strong circuit axiom with the fundamental circuits for I', while retaining the property that u is in the circuit; in the end we would obtain a circuit inside I', which is impossible. This contradiction proves the claim. □ We now show that G has a perfect matching by getting a contradiction. For each $u \in X$, let C_u be a circuit such that $C \cap I' \subseteq X$, u is the worst element of $C \cap X$ according to $\succ'$, and subject to that, the worst element in $C \cap Y$ according to $\succ'$ is best possible. Note that C_u exists, because the fundamental circuit of u for J' is a candidate, and each candidate circuit has an element in $C \cap Y$ by the previous claim.

Let $y(u)$ denote the worst element in $C_u \cap Y$ according to $\succ'$. Since $|Y| < |X|$, there exist $u_1 \in X$ and $u_2 \in X$ such that $y(u_1) = y(u_2)$; we may assume $u_1 \prec' u_2$. Let $y = y(u_1) = y(u_2)$; notice that $y \in C_{u_1} \cap C_{u_2}$ and $u_1 \in C_{u_1} \setminus C_{u_2}$. By the strong circuit axiom, we can obtain a circuit C such that $C \subseteq C_{u_1} \cup C_{u_2} - y$, and $u_1 \in C$. The existence of this circuit contradicts the choice of C_{u_1}. □

To prove Theorem 3, consider the perfect matching N given by Lemma 9. Then $w(N) \leq \sum_{v \in S'} \pi(v) = k$, so N has at most k edges uv for which $u \prec v$. This means that $\mathsf{vote}(J, I) \leq 2k - |I'|$, and therefore $\mathsf{vote}(I, J) + \mathsf{vote}(J, I) \leq 0$. This completes the proof of the theorem.

5 Algorithm

Here we describe Kamiyama's algorithm [14] in a generalized form. Given a pair of ordered matroids $M_i = (S, \mathcal{I}_i, \succ_i)$ $(i \in \{1,2\})$, we construct an extended instance $M_i^* = (S^*, \mathcal{I}_i^*, \succ_i^*)$ $(i \in \{1,2\})$ obtained by replacing each element with two parallel copies. Let the extended ground set be $S^* = \cup_{u \in S} \{x(u), y(u)\}$. The elements $x(u)$ and $y(u)$ are respectively called *x-copy* of u and *y-copy* of u. The independent set families are defined by

$$\mathcal{I}_i^* = \{\, I^* \subseteq S^* : \pi(I^*) \in \mathcal{I}_i,\ |I^* \cap \{x(u), y(u)\}| \leq 1\ (\forall u \in S)\,\},$$

where $\pi(I^*) = \{\, u \in S : I^* \cap \{x(u), y(u)\} \neq \emptyset\,\}$.

The linear order $\succ_i^*$ on S^* is defined as follows. In $\succ_1^*$, the x-copy of any element is preferred over the y-copy of any element, and the original preferences are preserved for the copies of the same type (e.g., $u \succ_1 v \Leftrightarrow x(u) \succ_1^* x(v),\ y(u) \succ_1^* y(v)$). In $\succ_2^*$, the roles of x and y are exchanged; the y-copies are preferred over the x-copies, and the original preferences are preserved for the copies of the same type. Kamiyama's algorithm is described as follows:

1. Find an (M_1^*, M_2^*)-kernel I^*.
2. Output $I := \pi(I^*)$.

Note that we can find a matroid kernel I^* in the first step efficiently by Fleiner's algorithm [8, 9].

Let I be the output of the algorithm. We show that I is super popular and largest among all weakly defendable common independent sets, which implies that I is a maximum popular common independent set by Corollary 7. To this end, we provide the following lemma.

Lemma 11 *For any $J \in \mathcal{I}_1 \cap \mathcal{I}_2$ and any weakly feasible pairings N_1 and N_2 for (I, J) with respect to matroids M_1 and M_2, respectively, we have $\mathsf{vote}_1(I, J, N_1) + \mathsf{vote}_2(I, J, N_2) \geq 0$. Moreover, if $|J| > |I|$, then $\mathsf{vote}_1(I, J, N_1) + \mathsf{vote}_2(I, J, N_2) > 0$.*

Before providing the proof of this lemma, we show that it easily implies the following theorems, which are our main results.

Theorem 12 *The output I of the algorithm is super popular and is largest among all weakly defendable common independent sets.*

Proof: The first claim of Lemma 11 implies $\mathsf{vote}_1^\bullet(I, J) + \mathsf{vote}_2^\bullet(I, J) \geq 0$ for any $J \in \mathcal{I}_1 \cap \mathcal{I}_2$, and hence I is super popular. By Corollary 7, then I is weakly defendable. By the second claim of Lemma 11, any common independent set $J \in \mathcal{I}_1 \cap \mathcal{I}_2$ larger than I satisfies $\mathsf{vote}_1^\bullet(I, J) + \mathsf{vote}_2^\bullet(I, J) > 0$, and hence J is not weakly defendable. Thus, I is a largest weakly defendable common independent set. □

Since we have Corollary 7 and the algorithm runs in polynomial time, the following theorem holds.

Theorem 13 *Given two ordered matroids $M_1 = (S, \mathcal{I}_1, \succ_1)$ and $M_2 = (S, \mathcal{I}_2, \succ_2)$, one can find a maximum popular common independent set in polynomial time.*

We now provide the proof of Lemma 11. It uses arguments similar to those used in Kavitha [15] and Kamiyama [14].

Proof of Lemma 11: Since each N_i is a weakly feasible pairing, $I - u + v \in \mathcal{I}_i$ for any $uv \in N_i$ and $I + v \in \mathcal{I}_i$ for any $v \in J \setminus I$ not covered by N_i. By the stability of I^*, any element in $J \setminus I$ is covered by N_1 or N_2. Consider the bipartite graph $G = (I \setminus J, J \setminus I; N_1 \cup N_2)$, which is decomposed into alternating paths, cycles, and isolated vertices in $I \setminus J$. For each path/cycle P, define its score as

$$\begin{aligned}\mathsf{score}(P) = &+ |\{\, uv \in P :\ uv \in N_i,\ u \succ_i v \text{ for some } i \in \{1,2\}\,\}| \\ &- |\{\, uv \in P :\ uv \in N_i,\ u \prec_i v \text{ for some } i \in \{1,2\}\,\}| \\ &+ 2(|P \cap (I \setminus J)| - |P \cap (J \setminus I)|),\end{aligned}$$

where we assume $u \in I \setminus J$ and $v \in I \setminus J$ and identify P with its edge set (resp., its vertex set) in the first and second terms (resp., in the third term). Note that $\mathsf{vote}_1(I, J, N_1) + \mathsf{vote}_2(I, J, N_2)$ equals the sum of the scores of all cycles/paths in G plus $2 \cdot \#\{\text{isolated vertices of } I \setminus J \text{ in } G\}$. Therefore, showing $\mathsf{score}(P) \geq 0$ for any path/cycle P completes the proof of the first claim of Lemma 11.

Let $u_0 v_1 u_1 v_2 u_2 \dots v_k u_k$ be the elements on P appearing in this order where $u_\ell \in I \setminus J$ and $v_\ell \in J \setminus I$ for each ℓ, and we set $u_0 = \emptyset$ if P starts at $J \setminus I$, we set $u_k = \emptyset$ if P ends at $J \setminus I$, and let $u_0 = u_k$ if P is a cycle. Without loss of generality, we assume $u_{\ell-1} v_\ell \in N_1$ and $u_\ell v_\ell \in N_2$ for each ℓ.

Consider the triple $u_{\ell-1} v_\ell u_\ell$ for $\ell = 1, 2, \dots, k$. Since I^* is stable, each of $x(v_\ell)$ and $y(v_\ell)$ should be dominated by I^* in M_1^* or M_2^*. Note that any x-copy (resp., y-copy) is preferred to any y-copy (resp., x-copy) in $\succ_1^*$ (resp., $\succ_2^*$) and that we have $u_{\ell-1} v_\ell \in N_1$ and $u_\ell v_\ell \in N_2$. Note also that $u_{\ell-1} = \emptyset$ (resp., $u_\ell = \emptyset$) implies that v_ℓ is uncovered in N_1 (resp., in N_2), and hence $I^* + y(v_\ell) \in \mathcal{I}_1$ (resp., $I^* + x(v_\ell) \in \mathcal{I}_2$). From these, we obtain the following conditions. Here, an element $u \in I \setminus J$ is called *x-type* (resp., *y-type*) if $I^* \cap \{x(u), y(u)\} = x(u)$ (resp., $y(u)$).

(a) If $u_{\ell-1}$ and u_ℓ are both x-type, then $u_{\ell-1} \succ_1 v_\ell$ or $u_\ell \succ_2 v_\ell$.

(b) If $u_{\ell-1}$ and u_ℓ are both y-type, then $u_{\ell-1} \succ_1 v_\ell$ or $u_\ell \succ_2 v_\ell$.

(c) If $u_{\ell-1}$ and u_ℓ are y-type and x-type, respectively, then $u_{\ell-1} \succ_1 v_\ell$ and $u_\ell \succ_2 v_\ell$.

(d) If $u_{\ell-1} = \emptyset$, then $u_\ell \succ_2 v_\ell$ and u_ℓ is y-type.

(e) If $u_\ell = \emptyset$, then $u_{\ell-1} \succ_1 v_\ell$ and $u_{\ell-1}$ is x-type.

The amount of votes obtained by the comparisons on $u_{\ell-1} v_\ell \in N_1$ and $u_\ell v_\ell \in N_2$ is nonnegative in all of the above cases, and in particular, it is 2 in case (c). This amount can be -2 only in the unlisted case, i.e., when $u_{\ell-1}$ and u_ℓ are x-type and y-type, respectively. Consider calculating the sum of the first two terms of $\mathsf{score}(P)$ by counting votes along P from u_0 to u_k. The value increases by 2 when u_ℓ turns from y-type to x-type, does not decrease when its type does not change, and decreases at most by 2 when u_ℓ turns from x-type to y-type. If P is a cycle, we can immediately obtain $\mathsf{score}(P) \geq 0$.

We then assume that P is a path. By the above arguments, the sum of the first two terms of $\mathsf{score}(P)$ is at least $2 \cdot (\#\{u_\ell \text{ turns from } y\text{-type to } x\text{-type}\} - \#\{u_\ell \text{ turns from } x\text{-type to } y\text{-type}\})$. The third term of $\mathsf{score}(P)$, i.e., $2(|P \cap (I \setminus J)| - |P \cap (J \setminus I)|)$, is $-2/0/2$ if both/either/none of u_0 and u_k is $\emptyset$. With the conditions (d) and (e), these imply $\mathsf{score}(P) \geq 0$.

Finally, we prove the second claim of the lemma. Suppose $|J| > |I|$. As we observed before, all elements in $J \setminus I$ are covered by $N_1 \cup N_2$. Since $|I \setminus J| < |J \setminus I|$, there exists a path $P = u_0 v_1 u_1 v_2 u_2 \dots v_k u_k$ in G that starts and ends at $J \setminus I$, i.e., $u_0 = u_k = \emptyset$. Then, the third term of $\mathsf{score}(P)$ is -2. By (d) and (e), we have $u_1 \succ_2 v_1$ and $u_{k-1} \succ_1 v_k$, from which we obtain 2 votes. From (d) and (e), we also obtain that u_1 is y-type while u_{k-1} is x-type, and hence $\#\{u_\ell \text{ turns from } y\text{-type to } x\text{-type}\}$ is strictly larger than $\#\{u_\ell \text{ turns from } x\text{-type to } y\text{-type}\}$. These imply that the sum of the first two terms of $\mathsf{score}(P)$ is at least 4. Thus, $\mathsf{score}(P) \geq 2 > 0$, and hence $\mathsf{vote}_1(I, J, N_1) + \mathsf{vote}_2(I, J, N_2) > 0$. □

6 Lexicographic Preferences

In the previous sections, we showed that finding a maximum popular matching in two-sided markets can be done in polynomial time, even if the two sides have arbitrary matroid constraints. However, our definition of popularity is not the only possible definition, and we may conceive other natural definitions of popularity for many-to-many settings. In this section we take a different approach and define popularity with respect to a much simpler voting rule, where the agents compare the two matchings/independent sets lexicographically. This means that they care mostly about their best element being as good as possible and with regard to that, their second best element being as good as possible, etc. This also implies that each agent has only one vote in the sense that they must choose a vote from the set $\{-1, 0, +1\}$ depending on which independent set they like better, similar to the one-to-one matching case. Note also that in this

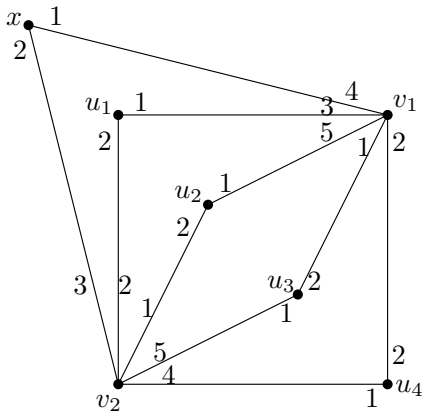


Figure 1: The instance given in Example 14. The numbers of the edges correspond to the preferences of the agents (smaller numbers are better).

model a smaller indepent set can be better than a much larger one, if the best element the agent obtains in the smaller one is better.

To make our proofs easier to follow, in this section we consider only partition matroids. As all our results are hardness results, they naturally extend to arbitrary matroids. In the case of partition matroids, where the voters correspond to exactly the connected components of the two matroids, we can model the instance with a bipartite graph $G = (U, W; E)$, where the vertices of $U \cup W$ correspond to the agents who vote and the edge set corresponds to the ground set of the matroids. We assume each agent $v \in U \cup W$ has a capacity $b(v)$ and a strict order $\succ^v$ over their adjacent edges, which we denote by $E(v)$.

Next, we define lexicographic popularity formally, for the case of the above b-matching problem. We call an edge set $M \subset E$ a *b-matching* if $|M \cap E(v)| \leq b(v)$ for all $v \in U \cup W$. Given two b-matchings M and M' in G and a vertex $v \in U \cup W$, we say that M is *lexicographically better than M' for v*, denoted by $M \succ^v_{\text{lex}} M'$, if $(E(v) \cap (M \cup M')) \setminus (M \cap M')$ is non-empty and the best element of this set according to the order $\succ^v$ is in M. (In the context of arbitrary matroids this would be generalized by saying that common independent set I is lexicographically better than I' for partition class U^i_j, $i \in \{1, 2\}$, if the best element of $(U^i_j \cap (I \cup I')) \setminus (I \cap I')$ according to the order $\succ_i$ is in I). Let $\mathsf{vote}^v_{\text{lex}}(M, M') \in \{-1, 0, +1\}$ denote the vote of agent v when comparing M' to M, that is, it is $+1$ if v lexicographically prefers M to M', -1 if v lexicographically prefers M' to M, and 0 otherwise. (Note that a vote can only be zero if $E(v) \cap M = E(v) \cap M'$, i.e. v obtains the same set of edges in both b-matchings). Let $\mathsf{vote}_{\text{lex}}(M, M') = \sum_{v \in U \cup W} \mathsf{vote}^v_{\text{lex}}(M, M')$ denote the sum of votes of all agents. We say that a b-matching M is *lexicographically popular*, if for any b-matching $M' \subset E$ it holds that $\mathsf{vote}_{\text{lex}}(M, M') \geq 0$. Otherwise, if $\mathsf{vote}_{\text{lex}}(M, M') < 0$, we say that M' *dominates* M. Clearly, it holds that $\mathsf{vote}_{\text{lex}}(M, M') = -\mathsf{vote}_{\text{lex}}(M', M)$ as exchangeability is symmetric for partition matroids.

In the standard one-to-one popular matching problem, lexicographic popularity and popularity coincide. However, when capacities can be larger, lexicographic popularity and popularity differ. First, we observe that in contrast to popular matchings, a lexicographically popular matching does not always exist.

Example 14 We give an example with no popular matching with respect to lexicographic preferences. We will also use this example as a gadget later and exploit its properties.

We have 7 agents, $x, u_1, \ldots, u_4$ on one side of the graph and v_1, v_2 on the other. The capacities of agents u_1 and u_4 are 1, the capacities of u_2, u_3, v_1, v_2 are 2 and the capacity of x is some number $q \geq 1$. (We do not fix q, because later in the hardness proofs we will use that no popular matching exists for any $q \geq 1$ in this example). The bipartite graph and the preferences are illustrated in Figure 1.

Suppose there is a lexicographically popular matching M. Then, (u_2, v_2) and (u_3, v_1) must be in M. Indeed, if one of them, say (u_2, v_2) is not in M, then u_2 is not saturated, so both v_2 and u_2 can improve

by adding (u_2, v_2) to M, and at most one agent gets worse (if v_2 is saturated and has to drop an edge).

Agent u_4 must be matched in M, because otherwise v_1 is either unsaturated or has a worse partner. So, by adding (u_4, v_1) to M and deleting the worse edge of v_1 if v_1 was saturated, we obtain a matching where v_1, u_4 both improve, and at most one agent gets worse. If u_4 is matched to v_2, then one of $\{x, u_1\}$ has to be totally unmatched in M. So, if agent v_2 drops u_4 and takes the free one of $\{x, u_1\}$, then only u_4 gets worse and two agents improve, contradicting the lexicographic popularity of M. So we have that $(u_4, v_1) \in M$.

We can see that u_1 must be matched by a similar argument as above, by replacing u_4 by u_1 and v_1 by v_2 in the argument. As we have already seen that v_1 is saturated by u_3 and u_4, (u_1, v_2) must be in M. We obtained that $M = \{(u_1, v_2), (u_2, v_2), (u_3, v_1), (u_4, v_1)\}$. However, M is dominated by the matching $M' = \{(u_1, v_1), (u_2, v_1), (u_3, v_2), (u_4, v_2)\}$, because u_1, u_2, u_3, u_4 all improve, and only v_1 and v_2 get worse.

Hence, we have shown that no lexicographically popular matching exists in this instance if x has capacity at least 1. However, if we add q dummy agents $d_1, d_2, \ldots, d_q$ who are only adjacent to x and x considers them the best, then there will be a unique lexicographically popular matching, namely $M = \{(u_1, v_1), (u_2, v_2), (u_3, v_1), (u_4, v_2)\} \cup \{(x, d_i) \mid i \in [q]\}$.

First we show that there can be no other lexicographically popular matching. By the same reasoning as before, $(u_2, v_2), (u_3, v_1)$ must be in M. Also, both u_1 and u_4 has to be matched. So the only other possibility for a lexicographically popular matching is $\{(u_1, v_2), (u_2, v_2), (u_3, v_1), (u_4, v_1)\} \cup \{(x, d_i) \mid i \in [q]\}$, but it is dominated by $\{(u_1, v_1), (u_2, v_1), (u_3, v_2), (u_4, v_2)\} \cup \{(x, d_i) \mid i \in [q]\}$.

Next we show that M is lexicographically popular. All agents other than u_2, u_3, v_1, v_2 are saturated and matched to their best partners, hence only these four can improve. As M is maximal, for any matching M' to dominate M there has to be an agent not from $\{u_2, u_3, v_1, v_2\}$ who gets worse, so the difference between the number of improving agents and the number of agents getting worse among u_2, u_3, v_1, v_2 must be at least 2. Let M' be a matching that dominates M. Agents u_2 or u_3 could only improve if v_1 or v_2 gets worse respectively. So, v_1 and v_2 must both improve, while u_2 and u_3 must not be worse off. This is only possible if v_1 gets u_4 and v_2 gets u_1. But then u_1 and u_4 both get worse, so $\mathsf{vote}_{\text{lex}}(M, M') \geq 0$, a contradiction.

With a counterexample in hand, we can show that deciding whether a lexicographically popular b-matching exists and verifying whether a b-matching is lexicographically popular are both hard.

Theorem 15 *It is coNP-hard to decide if a given instance $(G; \succ; b)$ admits a lexicographically popular b-matching. It is also coNP-complete to verify whether a given b-matching M is lexicographically popular. These hold even if each agent has capacity at most 3.*

The proof uses a reduction from the NP-complete Exact 3-Cover problem (X3C). Given an instance I of X3C, we can construct an instance I' of the b-matching problem such that I' has a unique candidate for lexicographically popular b-matching, and this candidate is lexicographically popular if and only if instance I does not have an exact 3-cover. The details of the construction and the proof of correctness can be found in the full version [6].

Proportional voting One might argue that agents should have voting weights proportional to their capacities in order to make the voting more fair. However, we can show that both the existence and verification problems remain hard even if all capacities are the same, using the following lemma. See the full version [6] for the proof.

Lemma 16 *For any instance $I = (G; \succ; b)$ with maximum capacity q, we can create an instance I', where every capacity is q, and there is a lexicographically popular b-matching in I' if and only if there is one in I. Furthermore, a b-matching M is lexicographically popular in I, if and only if by adding some fixed edges, the obtained b-matching M' is lexicographically popular in I'.*

References

[1] Péter Biró and Gergely Csáji. Strong core and Pareto-optimal solutions for the multiple partners matching problem under lexicographic preferences. *arXiv preprint arXiv:2202.05484*, 2022.

[2] Florian Brandl and Telikepalli Kavitha. Popular matchings with multiple partners. In *37th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, pages 19:1–19:15, 2018.

[3] Florian Brandl and Telikepalli Kavitha. Two problems in max-size popular matchings. *Algorithmica*, 81(7):2738–2764, 2019.

[4] Katarína Cechlárová, Pavlos Eirinakis, Tamás Fleiner, Dimitrios Magos, Ioannis Mourtos, and Eva Potpinková. Pareto optimality in many-to-many matching problems. *Discrete Optimization*, 14:160–169, 2014.

[5] le Marquis de Condorcet, Marie Jean Antoine Nicolas de Caritat. *Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix.* de l'Imprimerie Royale, 1785.

[6] Gergely Csáji, Tamás Király, and Yu Yokoi. Solving the Maximum Popular Matching Problem with Matroid Constraints. *arXiv preprint arXiv:2209.02195*, 2022.

[7] Gergely Csáji, Tamás Király, and Yu Yokoi. Approximation algorithms for matroidal and cardinal generalizations of stable matching. In *Symposium on Simplicity in Algorithms (SOSA)*, pages 103–113, 2023.

[8] Tamás Fleiner. A matroid generalization of the stable matching polytope. In *8th International Conference on Integer Programming and Combinatorial Optimization (IPCO)*, pages 105–114, 2001.

[9] Tamás Fleiner. A fixed-point approach to stable matchings and some applications. *Mathematics of Operations research*, 28(1):103–126, 2003.

[10] David Gale and Lloyd S Shapley. College admissions and the stability of marriage. *American Mathematical Monthly*, 69(1):9–15, 1962.

[11] Peter Gärdenfors. Match making: assignments based on bilateral preferences. *Behavioral Science*, 20(3):166–173, 1975.

[12] Sushmita Gupta, Pranabendu Misra, Saket Saurabh, and Meirav Zehavi. Popular matching in roommates setting is NP-hard. *ACM Transactions on Computation Theory*, 13(2):1–20, 2021.

[13] Chien-Chung Huang and Telikepalli Kavitha. Popular matchings in the stable marriage problem. In *International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 666–677, 2011.

[14] Naoyuki Kamiyama. Popular matchings with two-sided preference lists and matroid constraints. *Theoretical Computer Science*, 809:265–276, 2020.

[15] Telikepalli Kavitha. A size-popularity tradeoff in the stable marriage problem. *SIAM Journal on Computing*, 43(1):52–71, 2014.

[16] Meghana Nasre and Amit Rawat. Popularity in the generalized hospital residents setting. In *International Computer Science Symposium in Russia*, pages 245–259, 2017.

[17] Katarzyna Paluch. Popular and clan-popular b-matchings. *Theoretical Computer Science*, 544:3–13, 2014.

[18] Alexander Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*, volume 24. Springer, 2003.