Popular Arborescences and Their Matroid Generalization*

TELIKEPALLI KAVITHA, Tata Institute of Fundamental Research, India
KAZUHISA MAKINO, Kyoto University, Japan
ILDIKÓ SCHLOTTER, HUN-REN Centre for Economic and Regional Studies, Hungary
YU YOKOI, Institute of Science Tokyo, Japan

Consider a directed, rooted graph $G = (V \cup \{r\}, E)$ where each vertex in V has a partial order preference over its incoming edges. The preferences of a vertex naturally extend to preferences over arborescences rooted at r. We present a polynomial-time algorithm that decides whether a given input instance admits a popular arborescence, i.e., one for which there is no "more popular" arborescence.

In fact, our algorithm solves the more general popular common base problem in the intersection of two matroids: we are given an arbitrary matroid $M = (E, \mathcal{I})$ and a partition matroid M_{part} over E, where partition classes correspond to a set V of agents with $|V| = \text{rank}(M)$ and each agent has a partial order preference over its associated partition class; the problem asks for a common base of M and M_{part} such that there is no "more popular" common base. Our algorithm is combinatorial, and can be regarded as a primal–dual algorithm. It searches for a solution along with its dual certificate, a chain of subsets of E, witnessing its popularity. Our generalized results, expressed in terms of matroids, demonstrate that the identification of agents with vertices of the graph in the popular arborescence problem is not essential.

We also study the related popular common independent set problem. For the case with weak rankings, we formulate the popular common independent set polytope, and thus show that a minimum-cost popular common independent set can be computed efficiently. By contrast, we prove that it is NP-hard to compute a minimum-cost popular arborescence, even when rankings are strict.

CCS Concepts: • **Theory of computation → Design and analysis of algorithms**; • **Mathematics of computing → Matroids and greedoids**.

Additional Key Words and Phrases: matching under preferences, matroids, duality in linear programming

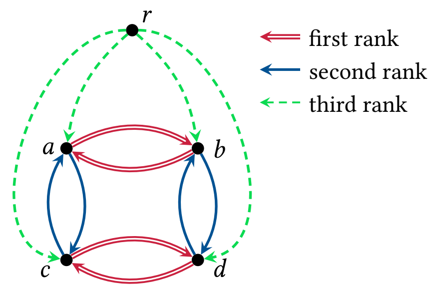
1 INTRODUCTION

Let $G = (V \cup \{r\}, E)$ be a directed graph where the vertex r (called the root) has no incoming edge. Every vertex $v \in V$ has a partial ordering $\succ_v$ (i.e., a preference relation that is irreflexive, antisymmetric and transitive) over the set $\delta(v)$ of its incoming edges, as in the following example from [22] where preference orders are strict rankings. Here $V = \{a, b, c, d\}$ and the preference orders of these four vertices on their incoming edges are as follows:

*A preliminary version has appeared in the 35th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2024) [24].

Authors' addresses: Telikepalli Kavitha, Tata Institute of Fundamental Research, Homi Bhabha Road, Mumbai, India, kavitha@tifr.res.in; Kazuhisa Makino, Kyoto University, Kitashirakawaoiwake-cho, Sakyo-ku, Kyoto, Japan, makino@kurims.kyoto-u.ac.jp; Ildikó Schlotter, HUN-REN Centre for Economic and Regional Studies, 1097 Tóth Kálmán u. 4., Budapest, Hungary, schlotter.ildiko@krtk.hun-ren.hu; Yu Yokoi, Institute of Science Tokyo, Ookayama, Meguro-ku, Tokyo, Japan, yokoi@comp.isct.ac.jp.

$$(b,a) \succ_a (c,a) \succ_a (r,a)$$
$$(a,b) \succ_b (d,b) \succ_b (r,b)$$
$$(d,c) \succ_c (a,c) \succ_c (r,c)$$
$$(c,d) \succ_d (b,d) \succ_d (r,d).$$



We are interested in computing an *optimal arborescence* rooted at r, where an arborescence is an acyclic subgraph in which each vertex, except one called the root, has a unique incoming edge; notice that all arborescences in our input graph G are rooted at r, as r has no incoming edge. Our notion of optimality is a function of the preferences $(\succ_v)_{v \in V}$ of vertices for their incoming edges.

Given any pair of arborescences A and A' in G, we say that $v \in V$ prefers A to A' if v prefers its incoming edge in A to its incoming edge in A', i.e., v prefers A to A' if $A(v) \succ_v A'(v)$ where $A(v)$ (respectively, $A'(v)$) is v's incoming edge in A (respectively, in A'). Let $\phi(A, A')$ be the number of vertices that prefer A to A'. We say that A is *more popular than* A' if $\phi(A, A') > \phi(A', A)$.

Definition 1.1. An arborescence A is popular if $\phi(A, A') \geq \phi(A', A)$ for all arborescences A'.

Our notion of optimality is popularity, in other words, we seek a popular arborescence A in G. So there is *no* arborescence more popular than A, thus A is maximal under the "more popular than" relation. The "more popular than" relation is not transitive and popular arborescences need not always exist.

Example 1.2. Consider the example from [22] depicted above. The arborescence $A = \{(r,a), (a,b), (a,c), (c,d)\}$ is not popular, since the arborescence $A' = \{(r,d), (d,c), (c,a), (a,b)\}$ is more popular. This is because a and c prefer A' to A, while d prefers A to A', and b is indifferent between A and A'. We can similarly obtain an arborescence $A'' = \{(r,b), (b,a), (b,d), (d,c)\}$ more popular than A'. It is easy to check that for any arborescence here, there is a more popular arborescence. Therefore this instance has no popular arborescence.

Consider the above instance without the edge (r,d). Vertex preferences are the same as in the earlier instance, except that vertex d has no third-choice edge. It can be shown that this instance has two popular arborescences: $A = \{(r,a), (a,b), (a,c), (c,d)\}$ and $A''' = \{(r,b), (b,a), (a,c), (c,d)\}$ (Appendix A has more details).

The popular arborescence problem. Given a directed graph G as described above, the popular arborescence problem is to determine if G admits a popular arborescence or not, and to find one, if so. The computational complexity of the popular arborescence problem was posed as an open problem at the Emléktábla workshop [26] in 2019 and the problem has remained open till now. Thus it is an intriguing open problem—aside from its mathematical interest and curiosity, it has applications in *liquid democracy*, which is a voting scheme that allows the delegation of votes. Each voter can delegate its vote to another voter; however, delegation cycles are forbidden. A popular arborescence represents a cycle-free delegation process that is stable, and every child of the root r casts a weighted vote on behalf of all its descendants.[1]

[1]A vertex v delegating its vote to u should be represented as the edge (v,u); however as said in [22], it will be more convenient to denote this delegation by (u,v) so as to be consistent with downward edges in an arborescence.

Liquid democracy has been used for internal decision making at *Google* [18] and political parties such as the German *Pirate Party* or the Swedish party *Demoex*. We refer to [32] for more details. We show the following result.

THEOREM 1.3. *Let $G = (V \cup \{r\}, E)$ be a directed graph where each $v \in V$ has a partial order over its incoming edges. There is a polynomial-time algorithm to solve the popular arborescence problem in G.*

1.1 Popular common base problem

The popular arborescence problem is a special case of the popular common base problem defined as follows. We are given two matroids, where one matroid is a partition matroid $M_{\text{part}} = (E, \mathcal{I}_{\text{part}})$ whose partition classes E_v are indexed by the elements v of a finite set V and the independent set family is defined by $\mathcal{I}_{\text{part}} = \{ S \subseteq E : |S \cap E_v| \leq 1 \text{ for all } v \in V \}$. The other matroid $M = (E, \mathcal{I})$ is an arbitrary matroid of rank $|V|$. Each $v \in V$ has a partial order $\succ_v$ over the partition class E_v associated with v. For any pair of common bases (i.e., common maximal independent sets) B and B' in the intersection of matroids M_{part} and M, we say that $v \in V$ prefers B to B' if v prefers the element in $B \cap E_v$ to the element in $B' \cap E_v$, i.e., $e \succ_v f$ where $B \cap E_v = \{e\}$ and $B' \cap E_v = \{f\}$. Let $\phi(B, B')$ be the number of elements in V that prefer B to B'.

Definition 1.4. A common base B of the matroids M_{part} and M is popular if $\phi(B, B') \geq \phi(B', B)$ for all common bases B'.

The task in the popular common base problem is to find a popular common base in the intersection of two such matroids or decide if no popular common base exists. Arborescences in a graph G are common bases in the intersection of the partition matroid on $E = \dot\bigcup_{v \in V} \delta(v)$ with the graphic matroid of G (where for any edge set $I \subseteq E$, $I \in \mathcal{I}$ if and only if I has no cycle in the underlying undirected graph). In fact, our algorithm solves the general popular common base problem.

THEOREM 1.5. *There exists a polynomial-time algorithm that computes a popular common base in the intersection of a partition matroid on $E = \dot\bigcup_{v \in V} E_v$ and any matroid $M = (E, \mathcal{I})$ of rank $|V|$, where each $v \in V$ has a partial order $\succ_v$ over elements in E_v, or decides that no such common base exists.*

In general, the matroid intersection need not admit common bases, and in such a case, an alternative is a largest common independent set that is popular among all largest common independent sets. This problem can be easily reduced to the popular common base problem (see Section 8). Furthermore, along with some simple reductions, we can use our popular common base algorithm to find a popular solution under certain constraints.

For example, we can find a common independent set that is popular subject to a size constraint (if a solution exists). We can further solve the problem under a category-wise size constraint: consider a setting of the popular arborescence problem and the liquid democracy application mentioned earlier, where the set V of voters is partitioned into categories, and for each category, there are lower and upper bounds on the number of voters who (roughly speaking) have an element in the chosen independent set belonging to them (see Section 8). This translates to setting lower and upper bounds on the number of representatives taken from each category so as to ensure that there is diversity among them.

Moreover, the popular common independent set problem which asks for a common independent set that is popular in the set of all common independent sets (of all sizes) in the matroid intersection can be easily reduced to the popular common base problem (see Section 4). Therefore, the following fact is obtained as a corollary to Theorem 1.5.

Corollary 1.6. *There exists a polynomial-time algorithm that computes a popular common independent set in the intersection of a partition matroid on $E = \dot\bigcup_{v\in V} E_v$ and any matroid $M = (E, \mathcal{I})$, where each $v \in V$ has a partial order $\succ_v$ over elements in E_v, or decides that no such common independent set exists.*

Popular common independent set polytope. Edmonds [12] proved that the intersection of the independent set polytopes of two matroids gives exactly the convex hull of the common independent sets of the two matroids—this is called the *matroid intersection polytope*; see also [33, Chapter 14.1]. If preferences are weak rankings, then we can give a formulation of an extension of the *popular common independent set polytope*, i.e., the convex hull of characteristic vectors of popular common independent sets in our matroid intersection.

Our formulation involves two auxiliary matroids that are used in our reduction from the popular common independent set problem to the popular common base problem. The common ground set of these auxiliary matroids is an extension of E with dummy elements. See Section 4 for the definitions of these matroids and a more precise formulation (Theorem 4.4) of the following statement.

Theorem 1.7. *If preferences are weak rankings, the popular common independent set polytope of a partition matroid on $E = \dot\bigcup_{v\in V} E_v$ and any matroid $M = (E, \mathcal{I})$ is a projection of a face of a matroid intersection polytope defined on a superset of E.*

By *projection*, we mean the deletion of variables corresponding to elements not in E. There are an exponential number of constraints in this formulation, however it admits an efficient separation oracle. As a consequence, when there is a function cost $: E \to \mathbb{R}$, a min-cost popular common independent set can be computed in polynomial time by optimizing over this polytope, assuming that preferences are weak rankings.

It follows from past work [21] that finding a min-cost popular common base—even with strict rankings—is NP-hard. Nevertheless, as shown here, finding a popular common base with forced or forbidden elements in an input instance with partial order preferences is polynomial-time solvable. This result allows us to recognize in polynomial time all those elements that are present in every popular common base and all those elements that are present in *no* popular common base.

Theorem 1.8. *Given a partition matroid on $E = \dot\bigcup_{v\in V} E_v$, where each $v \in V$ has a partial order $\succ_v$ over elements in E_v, and any matroid $M = (E, \mathcal{I})$ of rank $|V|$, along with a set $E^+ \subseteq E$ of forced elements and a set $E^- \subseteq E$ of forbidden elements, there is a polynomial-time algorithm to decide if there is a popular common base B with $E^+ \subseteq B$ and $E^- \cap B = \emptyset$ and to find one, if so.*

1.2 Related graph problems and our hardness results

In this section we discuss problems that are closely related to our starting problem—the popular arborescence problem. After explaining how these problems relate to the popular common base problem, we highlight some consequences of our algorithmic results presented in Section 1.1, and contrast them with NP-hardness results that outline the limits of tractability (Theorems 1.9 and 1.10).

Popular branchings. A special case of the popular arborescence problem is the popular branching problem. A branching is a directed forest in a digraph $G = (V, E)$ where each vertex has at most one incoming edge. Any branching in G can be viewed as an arborescence in an auxiliary graph obtained by augmenting G with a new vertex r as the root and adding the edge (r, v) for each $v \in V$ as the least-preferred incoming edge of v. So the problem of deciding whether the given instance G admits a popular branching or not reduces to the problem of deciding whether this auxiliary instance admits a popular arborescence or not. An efficient algorithm for this special case of the popular arborescence problem (where the root r is an in-neighbor of every $v \in V$) was given in [22].

The applications of popular branchings in liquid democracy were discussed in [22]. Every root in a popular branching B casts a weighted vote on behalf of all its descendants. However, in many real-world applications, not all agents would be willing to be representatives, i.e., to be roots in a branching. Thus, it cannot be assumed that *every* vertex is an out-neighbor of r, so it is only agents who are willing to be representatives that are out-neighbors of r in our instance. Therefore, the popular arborescence problem has to be solved in a general digraph $G = (V \cup \{r\}, E)$ rather than in one where every vertex is an out-neighbor of r.

Popular matchings and assignments. The notion of popularity has been extensively studied in the domain of bipartite matchings where vertices on one side of the graph have weak rankings (i.e., linear preference order with possible ties) over their neighbors. The popular matching problem is to decide if such a bipartite graph admits a *popular matching*, i.e., a matching M such that there is no matching more popular than M.

An efficient algorithm for the popular matching problem was given in 2007 [1]. The popular assignment problem was considered several years later in 2022 [21]. What is sought in this problem is a perfect matching that is popular within the set of perfect matchings—so the cardinality of the matching is more important than popularity here. It is easy to see that the popular assignment problem is a generalization of the popular matching problem; a simple reduction from the popular matching problem to the popular assignment problem can be shown by adding some dummy vertices (see [21, Section 2.3]). An efficient algorithm for the popular assignment problem was given in [21]. Note that assignments are common bases in the intersection of two partition matroids.

In view of the above connections, we can observe that all of the following problems fall in the framework of a popular common base (or common independent set) in the intersection of a partition matroid with another matroid:

(1) Popular matchings [1].
(2) Popular assignments [21].
(3) Popular branchings [22].
(4) Popular matchings with matroid constraints[2] [20].

Since Corollary 1.6 holds for partial order preferences, it generalizes the tractability result in [20] which assumes that preferences are weak rankings (note that the results in [20] are based on the paper [1], which in turn strongly relies on weak rankings). Observe that past work identified *agents* or elements with preferences with vertices in a graph. It follows from Theorem 1.5 that such an identification is not important at all. We exploit this property further to show that the following two new problems—popular colorful forest and popular colorful spanning tree—can also be solved efficiently by our algorithm. These are natural generalizations of the popular branching problem and the popular arborescence problem, respectively.

Popular colorful forests and popular colorful spanning trees. The input of the popular colorful forest problem is an undirected graph G where each edge has a color in $\{1, \ldots, n\}$. A forest F is *colorful* if each edge in F has a distinct color. Colorful forests are the common independent sets of the partition matroid defined by color classes and the graphic matroid of G. For each $i \in \{1, \ldots, n\}$, we assume that there is an agent i with a partial order $\succ_i$ over color i edges. Agent i prefers forest F

[2]This problem asks for a popular many-to-one matching in a bipartite graph $G = (A \cup B, E)$ where vertices in A have weak rankings and the vertices that get matched to each $b \in B$ must form an independent set in a matroid M_b.

to forest F' if either (i) F contains an edge colored i while F' has no edge colored i or (ii) both F and F' contain color i edges and i prefers the color i edge in F to the color i edge in F'.

A colorful forest F is popular if $\phi(F, F') \geq \phi(F', F)$ for all colorful forests F', where $\phi(F, F')$ is the number of agents that prefer F to F'. The popular colorful forest problem is to decide if a given graph G admits a popular colorful forest or not, and to find one, if so. The motivation here is to find an optimal *independent* network (cycles are forbidden) with diversity, i.e., there is at most one edge from each color class—as before, our definition of optimality is popularity. The popular branching problem is a special case of the popular colorful forest problem where all edges entering vertex i are colored i.

A *colorful spanning tree* is a colorful forest with exactly one component. In the popular colorful spanning tree problem, *connectivity* is more important than popularity, and we seek popularity within the set of colorful spanning trees rather than popularity within the set of all colorful forests.

Theorem 1.5 implies a polynomial-time algorithm to solve the popular colorful spanning tree problem and Corollary 1.6 implies a polynomial-time algorithm to solve the popular colorful forest problem. Recall that Theorem 1.7 gives a formulation of an extension of the popular colorful forest polytope when preferences are weak rankings. When preferences are weak rankings, note that such a polytope was already known for popular branchings [22] where it was also shown that it is NP-hard to find a min-cost popular branching with partial order preferences. It follows from Theorem 1.7 that for weak rankings, we can compute a min-cost popular colorful forest in polynomial time when there is a function $\mathsf{cost}: E \to \mathbb{R}$.

One may wonder if such a result also holds for the popular colorful spanning tree problem or at least the popular arborescence problem. Unfortunately, that is not the case as we show here.

Theorem 1.9. *Given an instance $G = (V \cup \{r\}, E)$ of the popular arborescence problem where each vertex has a strict ranking over its incoming edges along with a function* $\mathsf{cost}: E \to \{0, 1, \infty\}$, *it is* NP-*hard to compute a min-cost popular arborescence in G.*

A natural problem in instances that do not admit a popular arborescence is the relaxation of popularity to *near-popularity* or "low unpopularity". A standard measure of unpopularity is the *unpopularity margin* [28], defined for any arborescence A as $\mu(A) = \max_{A'} \phi(A', A) - \phi(A, A')$ where the maximum is taken over all arborescences A'. An arborescence A is popular if and only if $\mu(A) = 0$. Unfortunately, finding an arborescence with minimum unpopularity margin is NP-hard.

Theorem 1.10. *Given an instance $G = (V \cup \{r\}, E)$ of the popular arborescence problem where each vertex has a strict ranking over its incoming edges, together with an integer k, it is* NP-*complete to decide whether G contains an arborescence with unpopularity margin at most k.*

1.3 Background

The notion of popularity was introduced by Gärdenfors [15] in 1975 in bipartite graphs with two-sided strict preferences. In this model every stable matching [14] is popular, thus popular matchings always exist in this setting. When preferences are *one-sided*, popular matchings need not always exist. This is not very surprising given that popular solutions correspond to (weak) Condorcet winners [6, 29] and it is well-known in social choice theory that such a winner need not exist.

For the case when preferences are weak rankings, a combinatorial characterization of popular matchings was given in [1] and this yielded an efficient algorithm to solve the popular matching problem in this case. Note that the characterization in [1] does not generalize to partial order preferences, as argued in [23]. Several extensions of the popular matching problem have been considered such as random popular matchings [27], weighted voters [30], capacitated objects [34], popular mixed matchings [25], and popularity with matroid constraints [20]. We refer to [7] for a survey on results in popular matchings.

Popular spanning trees were studied in [8–10] where the incentive was to find a "socially best" spanning tree. However, in contrast to the popular colorful spanning tree problem, edges have no colors in their model and voters have rankings over the entire edge set. Many different ways to compare a pair of trees were studied here, and most of these led to hardness results.

Popular branchings, i.e., popular directed forests, in a directed graph (where each vertex has preferences as a partial order over its incoming edges) were studied in [22] where a polynomial-time algorithm was given for the popular branching problem. When preferences are weak rankings, polynomial-time algorithms for the min-cost popular branching problem and the k-unpopularity margin branching problem were shown in [22]; however these problems were shown to be NP-hard for partial order preferences. The popular branching problem where each vertex (i.e., voter) has a weight was considered in [31].

The popular assignment algorithm from [21] solves the popular maximum matching problem in a bipartite graph, and works for partial order preferences. It was also shown in [21] that the min-cost popular assignment problem is NP-hard, even for strict rankings.

Many combinatorial optimization problems can be expressed as (largest) common independent sets in the intersection of two matroids. Interestingly, constraining one of the two matroids in the matroid intersection to be a partition matroid is not really a restriction, because any matroid intersection can be reduced to the case where one matroid is a partition matroid (see [12, Claims 104–106]). We refer to [16, 33] for notes on matroid intersection and for the formulation of the matroid intersection polytope.

1.4 An overview of our algorithm

For a common base B, we can naturally define a weight function $\mathsf{wt}_B : E \longrightarrow \{-1, 0, 1\}$ such that for any common base B' we have $\mathsf{wt}_B(B') = \phi(B', B) - \phi(B, B')$. Then a popular common base B is a max-weight common base in the intersection of matroids M_{part} and M with respect to this function wt_B. Therefore, the popular common base problem is to find a common base B such that $\max_{B' \in \mathcal{B}} \mathsf{wt}_B(B') = \mathsf{wt}_B(B) = 0$ where $\mathcal{B}$ is the set of all common bases in the matroid intersection. This means that a popular common base B is an optimal solution to the linear program (LP) for the max-weight common base with weights given by wt_B.

Dual certificates. We show that every popular common base B has a dual certificate with a special structure, which corresponds to a *chain* $C = \{C_1, \ldots, C_p\}$ of subsets of E with $\emptyset \subsetneq C_1 \subsetneq \cdots \subsetneq C_p = E$ and $\text{span}(B \cap C_i) = C_i$ for all i.[3] Our algorithm to compute a popular common base is a search for such a chain C and a common base B. At a high level, this method is similar to the approach used in [21] for popular assignment, however our dual certificates are more complex than those in [21], and hence the steps in our algorithm (and its proof of correctness) become much more challenging.

Given a chain C of subsets of E, there is a polynomial-time algorithm to check if C corresponds to a dual certificate for some popular common base. It follows from dual feasibility and complementary slackness that C is a dual certificate if and only if for a certain subset $E(C) \subseteq E$, there exists a common base $B \subseteq E(C)$ such that $\text{span}(B \cap C_i) = C_i$ for all $C_i \in C$. If such a common base B exists, then it is easy to show that B is a popular common base with C as its dual certificate.

If such a common base $B \subseteq E(C)$ does not exist, then we need to update C. Since updating C changes $E(C)$, we now seek a common base $B \subseteq E(C)$ (for the new C) such that $\text{span}(B \cap C_i) = C_i$ for all i. If such a common base B does not exist yet again, then C is updated once more and so on.

[3] The set $\text{span}(B \cap C_i)$ is defined as $(B \cap C_i) \cup \{\, e \in E : (B \cap C_i) + e \text{ is not independent} \,\}$.

Note that updating C may increase $|C|$. When $|C|$ becomes larger than $|V|$, we claim that the input instance has *no* popular common base. Among other ideas, our technical novelty lies in the proof of this claim that is based on the strong exchange property of matroids.

We also show that a popular common independent set has a dual certificate $C = \{C, E\}$ of length at most 2. This leads to the polyhedral result given in Theorem 1.7.

Comparison with previous algorithms. In the case of popular arborescences, the problem of checking if a chain C corresponds to a dual certificate is the same as checking if a certain subgraph $G_C = (V \cup \{r\}, E(C))$ admits an arborescence A such that $\text{span}(A \cap C_i) = C_i$ for all $C_i \in C$. Our algorithm is different from the popular branching algorithm [22] that (loosely speaking) first finds a maximum branching on *best* edges and then augments this branching with *second best* edges entering certain vertices. Note that the popular arborescence problem regards reachability from the root r to be more important than popularity, thus a popular arborescence need not be a popular branching. However, in an instance with a popular branching that is also a popular arborescence, our algorithm will return such a popular branching.

Recall that there is a polynomial-time algorithm for finding a popular matching in a bipartite graph $G = (A \cup B, E)$ with matroid constraints on the B-side [20]. One may wonder if a polynomial-time algorithm to solve the popular common base problem can be obtained by identifying the entire set B into a single vertex. However this would require generalizing the algorithm in [20] to solve the popular *assignment* problem (rather than the popular matching problem) with matroid constraints in an instance with parallel edges. Recall that the popular matching problem is a special case of the popular assignment problem and fixing the size of the solution makes the problem involved. While popular matchings admit a simple characterization described by two edge sets [1], finding a popular perfect matching [21] requires a sophisticated primal–dual type algorithm. Furthermore, our algorithm for the popular common base problem works for preferences in partial orders, while the algorithm in [20] does not seem to generalize to that setting.[4]

Organization of the paper. The rest of the paper is organized as follows. Section 2 describes dual certificates for popular common bases. Section 3 presents the popular common base algorithm and its proof of correctness. In Section 4, we discuss popular common independent sets and their polytope. Section 5 provides the algorithm for the popular common base problem with forced/forbidden elements. Our hardness results are proved in Sections 6 and 7. Section 8 contains extensions and related results. Appendix A offers examples of executions of our algorithm.

2 DUAL CERTIFICATES

In this section we show that every popular common base has a special dual certificate—this will be crucial in designing our algorithm in Section 3.

In the popular common base problem, we are given a partition matroid M_{part} on a set $E = \dot\bigcup_{v \in V} E_v$, where V is the set of indices and each $v \in V$ is associated with a partial order $\succ_v$ over E_v, and an arbitrary matroid $M = (E, \mathcal{I})$ of rank $|V|$. For each $v \in V$ and each $e, f \in E_v$, we write $e \sim_v f$ to denote that v is indifferent between e and f, i.e., $e \not\succ_v f$ and $f \not\succ_v e$.

Given a common base B, there is a simple method (as shown in [22]) to check if B is popular or not. We need to check that $\phi(B, B') \geq \phi(B', B)$ for all common bases B' of M_{part} and M. For this, we will use the following function $\mathsf{wt}_B : E \to \{-1, 0, 1\}$.

[4]Solving the popular matching problem for partial order preferences requires a different approach [21] than the simple combinatorial characterization that forms the basis of the popular matching algorithm [1] which is generalized in [20].

For any $v \in V$, let $B(v)$ be the unique element in $B \cap E_v$. For any $v \in V$ and $e \in E_v$, let

$$\mathrm{wt}_B(e) = \begin{cases} 1 & \text{if } e \succ_v B(v) \quad (v \text{ prefers } e \text{ to } B(v)); \\ 0 & \text{if } e \sim_v B(v) \quad (v \text{ is indifferent between } e \text{ and } B(v)); \\ -1 & \text{if } e \prec_v B(v) \quad (v \text{ prefers } B(v) \text{ to } e). \end{cases}$$

It immediately follows from the definition of wt_B that we have $\mathrm{wt}_B(B') = \phi(B', B) - \phi(B, B')$ for any common base B'. Thus, B is popular if and only if every common base of M_{part} and M has weight at most 0, where weights on E are given by wt_B.

Consider the linear program problem LP1 below. The constraints of LP1 describe the face of the matroid intersection polytope corresponding to common bases. Here, $\mathrm{rank} : 2^E \to \mathbb{Z}_+$ is the rank function of the matroid $M = (E, \mathcal{I})$. That is, for each $S \subseteq E$, the value $\mathrm{rank}(S)$ is the size of a maximum independent subset of S.

$$\begin{aligned} \max \sum_{e \in E} \mathrm{wt}_B(e) \cdot x_e & & \text{(LP1)} \\ \text{s.t.} \sum_{e \in E_v} x_e = 1 & \quad \forall\, v \in V \\ \sum_{e \in S} x_e \le \mathrm{rank}(S) & \quad \forall\, S \subseteq E \\ x_e \ge 0 & \quad \forall\, e \in E. \end{aligned}$$

$$\begin{aligned} \min \sum_{S \subseteq E} \mathrm{rank}(S) \cdot y_S + \sum_{v \in V} \alpha_v & & \text{(LP2)} \\ \text{s.t.} \sum_{S : e \in S} y_S + \alpha_v \ge \mathrm{wt}_B(e) & \quad \forall\, e \in E_v, \forall v \in V \\ y_S \ge 0 & \quad \forall\, S \subseteq E. \end{aligned}$$

The feasible region of LP1 is the common base polytope of the matroids M_{part} and M. Hence LP1 is the max-weight common base LP with weights given by wt_B. The linear program LP2 is the dual LP in variables y_S and α_v where $S \subseteq E$ and $v \in V$.

The common base B is popular if and only if the optimal value of LP1 is at most 0, more precisely, if the optimal value is exactly 0, since $\mathrm{wt}_B(B) = 0$. Equivalently, B is popular if and only if the optimal value of LP2 is 0. We will now show that LP2 has an optimal solution with some special properties. For a popular common base B, a dual optimal solution that satisfies all these special properties (see Lemma 2.1) will be called a *dual certificate* for B.

The function $\mathrm{span} : 2^E \to 2^E$ of a matroid $(E, \mathcal{I})$ is defined as follows:

$$\mathrm{span}(S) = \{\, e \in E : \ \mathrm{rank}(S + e) = \mathrm{rank}(S) \,\} \quad \text{where } S \subseteq E.$$

In particular, if $S \in \mathcal{I}$, then $\mathrm{span}(S) = S \cup \{\, e \in E : S + e \notin \mathcal{I} \,\}$.

A *chain* $\mathcal{C}$ of length p is a collection of p distinct subsets of E such that for each two distinct sets $C, C' \in \mathcal{C}$, we have either $C \subsetneq C'$ or $C' \subsetneq C$. That is, a chain has the form $\mathcal{C} = \{C_1, C_2, \ldots, C_p\}$ where $C_1 \subsetneq C_2 \subsetneq \cdots \subsetneq C_p$.

Lemma 2.1 shows that LP2 always admits an optimal solution in the following special form. The proof is based on basic facts on matroid intersection and linear programming.

Lemma 2.1. *A common base B is popular if and only if there exists a feasible solution $(\vec{y}, \vec{\alpha})$ to LP2 such that $\sum_{S \subseteq E} \mathrm{rank}(S) \cdot y_S + \sum_{v \in V} \alpha_v = 0$ and properties 1–4 are satisfied:*

1. *$\vec{y}$ is integral and its support $\mathcal{C} := \{\, S \subseteq E : y_S > 0 \,\}$ is a chain.*
2. *Each $C \in \mathcal{C}$ satisfies $\mathrm{span}(B \cap C) = C$.*
3. *Every element in $\mathcal{C}$ is nonempty, and the maximal element in $\mathcal{C}$ is E.*
4. *For each $C \in \mathcal{C}$, we have $y_C = 1$. For each $v \in V$, we have $\alpha_v = -|\{\, C \in \mathcal{C} : B(v) \in C \,\}|$.*

Proof. The optimal value of LP1 is at least 0 since $\mathrm{wt}_B(B) = 0$. Therefore, if there exists a feasible solution $(\vec{y}, \vec{\alpha})$ to LP2 whose objective value is 0, then $(\vec{y}, \vec{\alpha})$ is an optimal solution to LP2 and the optimal value of LP1 is also 0, and hence B is a popular common base of M_{part} and M.

If B is a popular common base, then the optimal value of LP2 is 0. We will now show that there always exists an optimal solution $(\vec{y}, \vec{\alpha})$ to LP2 that satisfies properties 1–4.

1. It is a well-known fact on matroid intersection (see, e.g., [33, Theorem 41.12]) that there exists an integral optimal solution to LP2 such that the support of the dual variables corresponding to the matroid M is a chain, which means property 1. Here, we give a proof to make the paper self-contained.

Among all the optimal solutions to LP2, let $(\vec{y}, \vec{\alpha})$ be the one that minimizes $\sum_{S \subseteq E} y_S \cdot |S| \cdot |E \setminus S|$. Suppose to the contrary that the support of $\vec{y}$ is not a chain. Then there exist $X, Y \subseteq E$ such that $X \setminus Y \neq \emptyset$, $Y \setminus X \neq \emptyset$, $y_X > 0$, and $y_Y > 0$. Set $\delta = \min\{y_X, y_Y\}$, decrease y_X and y_Y by δ, increase $y_{X \cup Y}$ and $y_{X \cap Y}$ by δ, and denote the resultant vector by $\vec{z}$. Then, $\sum_{S: e \in S} y_S = \sum_{S: e \in S} z_S$ for any $e \in E$, and hence $(\vec{z}, \vec{\alpha})$ is feasible in LP2. Since $\mathrm{rank}(X) + \mathrm{rank}(Y) \geq \mathrm{rank}(X \cup Y) + \mathrm{rank}(X \cap Y)$ by the submodularity of the matroid rank function, the objective value of $(\vec{z}, \vec{\alpha})$ is not smaller than that of $(\vec{y}, \vec{\alpha})$, and hence $(\vec{z}, \vec{\alpha})$ is optimal. Since

$$\sum_{S \subseteq E} z_S \cdot |S| \cdot |E \setminus S| - \sum_{S \subseteq E} y_S \cdot |S| \cdot |E \setminus S| = \delta \cdot (|X|^2 + |Y|^2 - |X \cup Y|^2 - |X \cap Y|^2) < 0,$$

this contradicts the choice of $(\vec{y}, \vec{\alpha})$.

Thus, LP2 has an optimal solution $(\vec{y}^*, \vec{\alpha}^*)$ such that the support of $\vec{y}^*$ is a chain. Let $\mathcal{C}$ be the support of such $\vec{y}^*$ and LP2′ be a variant of LP2 such that variables $\vec{y}$ are defined as $\{y_C\}_{C \in \mathcal{C}}$ instead of $\{y_S\}_{S \subseteq E}$. Then, optimal solutions to LP2′ are also optimal solutions to LP2. Let $\mathcal{C}' = \{E_v\}_{v \in V}$. Then, both $\mathcal{C}$ and $\mathcal{C}'$ are laminar families on E. Let M be the $E \times (\mathcal{C} \cup \mathcal{C}')$ incidence matrix, i.e., $M_{e,C} = 1$ if $e \in C$ and 0 otherwise. Since the incidence matrix of the union of two laminar families is totally unimodular [12] (see also [33, Theorem 41.11]), so is M. This implies that LP2′ has an integral optimal solution, which is a solution to LP2 satisfying property 1.

2. Among all the optimal solutions to LP2 that satisfy property 1, let $(\vec{y}, \vec{\alpha})$ be the one that minimizes $\sum_{C \in \mathcal{C}} |\mathrm{span}(C) \setminus C|$, where $\mathcal{C}$ is the support of $\vec{y}$. We claim that $\mathrm{span}(B \cap C) = C$ holds for all $C \in \mathcal{C}$. Observe that each $C \in \mathcal{C}$ satisfies $y_C > 0$, and hence complementary slackness implies that the characteristic vector $\vec{x}$ of B satisfies $\sum_{e \in C} x_e = \mathrm{rank}(C)$, i.e., $|B \cap C| = \mathrm{rank}(C)$. Therefore, to obtain $\mathrm{span}(B \cap C) = C$ for all $C \in \mathcal{C}$, it suffices to show $\mathrm{span}(C) = C$ for all $C \in \mathcal{C}$. Suppose to the contrary that it does not hold. Then there exists at least one $C \in \mathcal{C}$ with $\mathrm{span}(C) \neq C$. Among all such C, let $C^* \in \mathcal{C}$ be the maximal one.

Define $\vec{z}$ as follows: (i) $z_{\mathrm{span}(C^*)} = y_{\mathrm{span}(C^*)} + y_{C^*}$, (ii) $z_{C^*} = 0$, and (iii) $z_S = y_S$ for all other $S \subseteq E$. Then $\mathcal{C}' = (\mathcal{C} \setminus \{C^*\}) \cup \{\mathrm{span}(C^*)\}$ is the support of $\vec{z}$. Note that $\mathcal{C}'$ is again a chain because any $C \in \mathcal{C}$ with $C^* \subsetneq C$ satisfies $\mathrm{span}(C) = C$ by the choice of C^*, hence $\mathrm{span}(C^*) \subseteq \mathrm{span}(C) = C$.

Observe that $(\vec{z}, \vec{\alpha})$ is a feasible solution to LP2. Moreover, since $\mathrm{rank}(C^*) = \mathrm{rank}(\mathrm{span}(C^*))$, it does not change the objective value. Thus $(\vec{z}, \vec{\alpha})$ is an optimal solution to LP2 that satisfies property 1 and $\sum_{C \in \mathcal{C}'} |\mathrm{span}(C) \setminus C| < \sum_{C \in \mathcal{C}} |\mathrm{span}(C) \setminus C|$. This contradicts the choice of $(\vec{y}, \vec{\alpha})$.

3. Suppose $(\vec{y}, \vec{\alpha})$ satisfies properties 1–2 but not property 3. If $\emptyset \in \mathcal{C}$, then modify $\vec{y}$ by setting $y_\emptyset = 0$. Then $\emptyset$ is removed from $\mathcal{C}$. This does not change the objective value and does not violate feasibility constraints.

If $E \notin \mathcal{C}$, then add E to $\mathcal{C}$ and modify $(\vec{y}, \vec{\alpha})$ by (i) setting $y_E = 1$ and (ii) decreasing every α_v value by 1. Since $\mathrm{rank}(E) = |V|$, the objective value does not change. Also, all constraints in LP2 are preserved. Hence the new solution satisfies properties 1–3.

4. Among all the optimal solutions to LP2 that satisfy properties 1–3, let $(\vec{y}, \vec{\alpha})$ be the one that minimizes $\sum_{S \subseteq E} y_S$ and let $\mathcal{C}$ be the support of $\vec{y}$. Note that $\alpha_v = -\sum_{C \in \mathcal{C}: B(v) \in C} y_C$ holds for any $v \in V$ by complementary slackness (observe that $x_{B(v)} > 0$ for B's characteristic vector $\vec{x}$).

Suppose $y_{C^*} \geq 2$ for some $C^* \in \mathcal{C}$. Define $(\vec{z}, \vec{\beta})$ as follows: $z_{C*} = y_{C^*} - 1$ and $z_S = y_S$ for every other $S \subseteq E$. For any $v \in V$, let $\beta_v = -\sum_{C \in \mathcal{C}: B(v) \in C} z_C$. We will show below that $(\vec{z}, \vec{\beta})$ is a feasible solution to LP2. Let us first see what is the objective value attained by $(\vec{z}, \vec{\beta})$.

This value is $\sum_{C \in \mathcal{C}} \mathrm{rank}(C) \cdot z_C + \sum_{v \in V} \beta_v$. When compared to $\sum_{C \in \mathcal{C}} \mathrm{rank}(C) \cdot y_C + \sum_{v \in V} \alpha_v$, the first term has decreased by $\mathrm{rank}(C^*)$ and the second term has increased by $|\{ v \in V : B(v) \in C^* \}| = |B \cap C^*| \leq \mathrm{rank}(C^*)$. Thus the objective value does not increase.

We will now show that $(\vec{z}, \vec{\beta})$ is a feasible solution to LP2, that is, $\sum_{C \in \mathcal{C}: e \in C} z_C + \beta_v \geq \mathrm{wt}_B(e)$ for each $e \in E_v$, $v \in V$. Since $(\vec{y}, \vec{\alpha})$ is feasible and the first term $\sum_{C \in \mathcal{C}: e \in C} z_C$ decreases by at most 1 and the second term $\beta_v = -\sum_{C \in \mathcal{C}: B(v) \in C} z_C$ never decreases, the only case we need to worry about is when the first term decreases and the second term does not increase. This implies that $e \in C^*$ and $B(v) \notin C^*$; hence

$$\sum_{C \in \mathcal{C}: e \in C} z_C + \beta_v = \sum_{C \in \mathcal{C}: e \in C} z_C - \sum_{C \in \mathcal{C}: B(v) \in C} z_C \geq z_{C*} \geq 1 \geq \mathrm{wt}_B(e).$$

Thus $(\vec{z}, \vec{\beta})$ is a feasible solution to LP2; furthermore, it is an optimal solution to LP2. Since we have $\sum_{S \subseteq E} z_S < \sum_{S \subseteq E} y_S$, this contradicts the choice of $(\vec{y}, \vec{\alpha})$.

Thus, we have shown that $(\vec{y}, \vec{\alpha})$ satisfies properties 1–3 and $y_C = 1$ for all $C \in \mathcal{C}$. Since we have $\alpha_v = -\sum_{C \in \mathcal{C}: B(v) \in C} y_C$, it follows that $\alpha_v = -|\{ C \in \mathcal{C} : B(v) \in C \}|$ for each $v \in V$. □

For any chain $\mathcal{C}$, we will now define a subset $E(\mathcal{C})$ of the ground set E that will be used in our algorithm. The construction of $E(\mathcal{C})$ is inspired by the construction of an analogous edge subset in the popular assignment algorithm [21].

For a chain $\mathcal{C} = \{C_1, C_2, \cdots, C_p\}$ with $\emptyset \subsetneq C_1 \subsetneq \cdots \subsetneq C_p = E$, define

$$\mathrm{lev}_{\mathcal{C}}(e) = \text{the index } i \text{ such that } e \in C_i \setminus C_{i-1} \quad \text{for any } e \in E,$$
$$\mathrm{lev}^*_{\mathcal{C}}(v) = \max \{ \mathrm{lev}_{\mathcal{C}}(e) : e \in E_v \} \qquad \text{for any } v \in V,$$

where we let $C_0 = \emptyset$. Thus every element $e \in E$ has a *level* in $\{1, \ldots, p\}$ associated with it, which is the minimum subscript i such that $e \in C_i$ (where $C_i \in \mathcal{C}$). Furthermore, each $v \in V$ has a $\mathrm{lev}^*_{\mathcal{C}}$-value which is the highest level of any element in E_v.

Define $E(\mathcal{C}) \subseteq E$ as follows. For each $v \in V$, an element $e \in E_v$ belongs to $E(\mathcal{C})$ if one of the following two conditions holds:

- $\mathrm{lev}_{\mathcal{C}}(e) = \mathrm{lev}^*_{\mathcal{C}}(v)$ and there is no element $e' \in E_v$ such that $\mathrm{lev}_{\mathcal{C}}(e') = \mathrm{lev}^*_{\mathcal{C}}(v)$ and $e' \succ_v e$;
- $\mathrm{lev}_{\mathcal{C}}(e) = \mathrm{lev}^*_{\mathcal{C}}(v) - 1$ and there is no element $e' \in E_v$ such that $\mathrm{lev}_{\mathcal{C}}(e') = \mathrm{lev}^*_{\mathcal{C}}(v) - 1$ and $e' \succ_v e$, and moreover, $e \succ_v f$ for every $f \in E_v$ with $\mathrm{lev}_{\mathcal{C}}(f) = \mathrm{lev}^*_{\mathcal{C}}(v)$.

In other words, $e \in E_v$ belongs to $E(\mathcal{C})$ if either (i) e is a maximal element in E_v with respect to $\succ_v$ among those at level $\mathrm{lev}^*_{\mathcal{C}}(v)$ or (ii) e is a maximal element in E_v among those at level $\mathrm{lev}^*_{\mathcal{C}}(v) - 1$ and v strictly prefers e to all elements at level $\mathrm{lev}^*_{\mathcal{C}}(v)$. From Lemma 2.1, we obtain the following useful characterization of popular common bases.

Lemma 2.2. *A common base B is popular if and only if there exists a chain $\mathcal{C} = \{C_1, \ldots, C_p\}$ such that $\emptyset \subsetneq C_1 \subsetneq \cdots \subsetneq C_p = E$, $B \subseteq E(\mathcal{C})$, and* $\mathrm{span}(B \cap C_i) = C_i$ *for all $C_i \in \mathcal{C}$.*

The proof is given below. Recall that for a popular common base B, we defined its *dual certificate* as a dual optimal solution $(\vec{y}, \vec{\alpha})$ to LP2 that satisfies properties 1–4 in Lemma 2.1. As shown in the proof of Lemma 2.2, we can obtain such a solution $(\vec{y}, \vec{\alpha})$ from a chain satisfying the properties in

Lemma 2.2. We therefore will also use the term *dual certificate* to refer to a chain as described in Lemma 2.2.

Proof of Lemma 2.2. We first show the existence of a desired chain $\mathcal{C}$ for a popular common base B. Since B is popular, we know from Lemma 2.1 that there exists an optimal solution $(\vec{y}, \vec{\alpha})$ to LP2 such that properties 1–4 hold, where $\mathcal{C}$ is the support of $\vec{y}$. Since the properties $\emptyset \subsetneq C_1 \subsetneq \cdots \subsetneq C_p = E$ and $\mathrm{span}(B \cap C_i) = C_i$ $(\forall C_i \in \mathcal{C})$ directly follow from properties 3 and 2, respectively, it remains to show that $B \subseteq E(\mathcal{C})$.

Since $(\vec{y}, \vec{\alpha})$ is a feasible solution of LP2, we have $\sum_{S: e \in S} y_S + \alpha_v \geq \mathrm{wt}_B(e)$ for every $e \in E_v$ with $v \in V$. By property 4, the left-hand side can be expressed as

$$
\begin{aligned}
|\{ C_i \in \mathcal{C} : e \in C_i \}| - |\{ C_i \in \mathcal{C} : B(v) \in C_i \}| &= (p - \mathsf{lev}_{\mathcal{C}}(e) + 1) - (p - \mathsf{lev}_{\mathcal{C}}(B(v)) + 1) \\
&= \mathsf{lev}_{\mathcal{C}}(B(v)) - \mathsf{lev}_{\mathcal{C}}(e).
\end{aligned}
$$

Thus it is equivalent to the condition that for every $e \in E_v$:

$$
\mathsf{lev}_{\mathcal{C}}(B(v)) - \mathsf{lev}_{\mathcal{C}}(e) \geq \mathrm{wt}_B(e) = \begin{cases} 1 & \text{if } e \succ_v B(v); \\ 0 & \text{if } e \sim_v B(v); \\ -1 & \text{if } e \prec_v B(v). \end{cases} \tag{1}
$$

In particular, this holds for an element e' with $\mathsf{lev}_{\mathcal{C}}(e') = \mathsf{lev}^*_{\mathcal{C}}(v)$, and hence we have $\mathsf{lev}_{\mathcal{C}}(B(v)) \geq \mathsf{lev}^*_{\mathcal{C}}(v) - 1$. Since $\mathsf{lev}_{\mathcal{C}}(B(v)) \leq \mathsf{lev}^*_{\mathcal{C}}(v)$ by $B(v) \in E_v$, $\mathsf{lev}_{\mathcal{C}}(B(v))$ is either $\mathsf{lev}^*_{\mathcal{C}}(v)$ or $\mathsf{lev}^*_{\mathcal{C}}(v) - 1$.

- If $\mathsf{lev}_{\mathcal{C}}(B(v)) = \mathsf{lev}^*_{\mathcal{C}}(v)$, then for any $e \in E_v$ with $\mathsf{lev}_{\mathcal{C}}(e) = \mathsf{lev}^*_{\mathcal{C}}(v)$, the left-hand side of (1) is 0, and hence it must be the case that either $B(v) \succ_v e$ or $B(v) \sim_v e$. Hence $B(v)$ is a maximal element in $\{ e \in E_v : \mathsf{lev}_{\mathcal{C}}(e) = \mathsf{lev}^*_{\mathcal{C}}(v) \}$ with respect to $\succ_v$.
- If $\mathsf{lev}_{\mathcal{C}}(B(v)) = \mathsf{lev}^*_{\mathcal{C}}(v) - 1$, then we can similarly show that $B(v)$ is a maximal element in the set $\{ e \in E_v : \mathsf{lev}_{\mathcal{C}}(e) = \mathsf{lev}^*_{\mathcal{C}}(v) - 1 \}$ with respect to $\succ_v$. Furthermore, in this case, for any $e \in E_v$ with $\mathsf{lev}_{\mathcal{C}}(e) = \mathsf{lev}^*_{\mathcal{C}}(v)$, the left-hand side of (1) is -1, and hence $B(v) \succ_v e$ must hold.

Therefore, in either case, we have $B(v) \in E(\mathcal{C})$, which implies that $B \subseteq E(\mathcal{C})$.

For the converse, suppose that $\mathcal{C} = \{C_1, \ldots, C_p\}$ is a chain such that $\emptyset \subsetneq C_1 \subsetneq \cdots \subsetneq C_p = E$, $B \subseteq E(\mathcal{C})$, and $\mathrm{span}(B \cap C_i) = C_i$ for all $C_i \in \mathcal{C}$. Define $\vec{y}$ by $y_{C_i} = 1$ for every $C_i \in \mathcal{C}$ and $y_S = 0$ for all $S \in 2^S \setminus \mathcal{C}$. We also define $\vec{\alpha}$ by $\alpha_v = -|\{ C \in \mathcal{C} : B(v) \in C \}|$ for any $v \in V$. Then $(\vec{y}, \vec{\alpha})$ satisfies properties 1–4 given in Lemma 2.1, and these properties also imply that the objective value is 0. Thus it is enough to show that $(\vec{y}, \vec{\alpha})$ is a feasible solution to LP2, because it implies that B is a popular common base by Lemma 2.1. Observe that constraint (1) is satisfied for every $v \in V$ and $e \in E_v$, which follows from $B \subseteq E(\mathcal{C})$. Since it is equivalent to the constraint in LP2 for $v \in V$ and $e \in E_v$, the proof is completed. □

3 OUR ALGORITHM

We now present our main result. The popular common base algorithm seeks to construct a common base B along with its dual certificate $\mathcal{C} = \{C_1, \ldots, C_p\}$, which is a chain satisfying (i) $\emptyset \subsetneq C_1 \subsetneq \cdots \subsetneq C_p = E$, (ii) $B \subseteq E(\mathcal{C})$, and (iii) $\mathrm{span}(B \cap C_i) = C_i$ for all $C_i \in \mathcal{C}$.

- The existence of such a chain $\mathcal{C}$ means that B is popular by Lemma 2.2.
- Since a popular common base need not always exist, the algorithm also needs to detect when a solution does not exist.

The algorithm starts with the chain $\mathcal{C} = \{E\}$ and repeatedly updates it. It always maintains $\mathcal{C}$ as a multichain, where a collection $\mathcal{C} = \{C_1, \cdots, C_p\}$ of indexed subsets of E is called a *multichain* if $C_1 \subseteq \cdots \subseteq C_p$. Note that it is a chain if all the inclusions are strict. We will use the notations $\mathsf{lev}_{\mathcal{C}}$, $\mathsf{lev}^*_{\mathcal{C}}$, and $E(\mathcal{C})$ also for multichains, which are defined in the same manner as for chains.

During the algorithm's execution, $\mathcal{C} = \{C_1, \ldots, C_p\}$ is always a multichain with $C_p = E$ and $\mathrm{span}(C_i) = C_i$ for all $C_i \in \mathcal{C}$. Note that when $\mathrm{span}(C_i) = C_i$ holds, for any set $B \subseteq E$, the condition $\mathrm{span}(B \cap C_i) = C_i$ in (iii) above is equivalent to $|B \cap C_i| = \mathrm{rank}(C_i)$. Furthermore, as explained later, any multichain can be modified to a chain that satisfies condition (i) preserving the remaining conditions (ii) and (iii). Therefore, we can obtain a desired chain if $|B \cap C_i| = \mathrm{rank}(C_i)$ is attained for all $C_i \in \mathcal{C}$ for some common base $B \subseteq E(\mathcal{C})$ in the algorithm.

Lex-maximal common independent set. In order to determine the existence of a common base $B \subseteq E(\mathcal{C})$ that satisfies $|B \cap C_i| = \mathrm{rank}(C_i)$ for all members $C_i \in \mathcal{C}$ of the chain, the algorithm computes a *lex-maximal* common independent set I in $E(\mathcal{C})$. That is, I is a common independent set whose p-tuple $(|I \cap C_1|, \ldots, |I \cap C_p|)$ is lexicographically maximum among all common independent sets in $E(\mathcal{C})$. If $(|I \cap C_1|, \ldots, |I \cap C_p|) = (\mathrm{rank}(C_1), \ldots, \mathrm{rank}(C_p))$, then we can show that I is a popular common base[5]; otherwise the multichain $\mathcal{C}$ is updated. We describe the algorithm as Algorithm 1; recall that $\mathrm{rank}(E) = |V|$.

Algorithm 1 The popular common base algorithm

1: Initialize $p = 1$ and $C_1 = E$. ▷ Initially we set $\mathcal{C} = \{E\}$.
2: **while** $p \leq |V|$ **do**
3: Compute the subset $E(\mathcal{C})$ of E from the current multichain $\mathcal{C}$.
4: Find a common independent set I that lexicographically maximizes $(|I \cap C_1|, \ldots, |I \cap C_p|)$ subject to $I \subseteq E(\mathcal{C})$.
5: **if** $|I \cap C_i| = \mathrm{rank}(C_i)$ for every $i = 1, \ldots, p$ **then** return I.
6: Let k be the minimum index such that $|I \cap C_k| < \mathrm{rank}(C_k)$.
7: Update $C_k \leftarrow \mathrm{span}(I \cap C_k)$.
8: **if** $k = p$ **then** $p \leftarrow p + 1$, $C_p \leftarrow E$, and $\mathcal{C} \leftarrow \mathcal{C} \cup \{C_p\}$.
9: Return "The input instance has no popular common base."

We include some example instances of the popular arborescence problem in Appendix A to illustrate the working of Algorithm 1 on the arborescence setting.

The following observation is important.

Observation 3.1. *During the execution of Algorithm 1, $\mathcal{C}$ is always a multichain and* $\mathrm{span}(C_i) = C_i$ *holds for all* $C_i \in \mathcal{C}$.

Proof. When C_k is updated, it becomes smaller, but the inclusion $C_{k-1} \subseteq C_k$ is preserved. Indeed, since $|I \cap C_{k-1}| = \mathrm{rank}(C_{k-1})$ by the choice of k, we have $C_{k-1} \subseteq \mathrm{span}(I \cap C_{k-1}) \subseteq \mathrm{span}(I \cap C_k)$, for the set C_k before the update. Hence the updated value for C_k, i.e., $\mathrm{span}(I \cap C_k)$, is still a superset of C_{k-1}, and thus $\mathcal{C}$ remains a multichain.

Since any $C_i \in \mathcal{C}$ is defined in the form $\mathrm{span}(X)$ for some $X \subseteq E$ (note that $E = \mathrm{span}(E)$) and $\mathrm{span}(\mathrm{span}(X)) = \mathrm{span}(X)$ holds in general, we have $\mathrm{span}(C_i) = C_i$. □

Running time of the algorithm. Before showing the correctness of the algorithm, here we briefly discuss the time complexity of Algorithm 1 on an instance $\mathcal{J}$ with $|E| = m$ and $|V| = n$. We assume that the partial order $\succ_v$ over E_v associated with each $v \in V$ is represented by a Hasse diagram, given as part of $\mathcal{J}$. Then Line 3 can be implemented in $O(|\mathcal{J}|)$ time. Note that a Hasse diagram

[5]Observe that the common independent set I will be a *common base* since $|I \cap E| = |I \cap C_p| = \mathrm{rank}(C_p) = \mathrm{rank}(E) = |V|$.

over E_v may require $\Theta(|E_v|^2)$ space, so the total size of Hasse diagrams may add up to $\Theta(m^2)$. Line 4 can be executed by solving an instance of the weighted matroid intersection problem, for which various efficient algorithms are known (see, e.g., [13, 19, 33]). Here, we assume that the matroid $M = (E, \mathcal{I})$ is given by an independence oracle or a rank function oracle. Lines 5–8 require $O(m)$ time. To bound the number of iterations of the while loop, consider the value $\sum_{i=1}^{n} \mathrm{rank}(C_i)$, where we let $\mathrm{rank}(C_i) = n$ for each C_i that is undefined. This value is n^2 at the beginning and decreases in every iteration. Thus, the number of iterations is at most n^2. These imply that Algorithm 1 can be implemented in polynomial time.

In the case of the popular arborescence problem, our input is a directed rooted graph with vertex preferences and we can provide a more precise complexity analysis. In this case, Line 4 can be done by a max-weight branching algorithm [2, 5, 11], which can be implemented to run in $O(m \log n)$ by Tarjan's algorithm [35]. Then, Algorithm 1 runs in $O(n^2(m \log n + |\mathcal{J}|))$ time, where the total size of Hasse diagrams of input preferences may add up to $\Theta(n^3)$ in this case. This running time can be further simplified to $O(n^2 m \log n)$ if preferences are weak rankings, which can be simply represented as preference lists.

Correctness of the algorithm. Suppose that a common independent set I is returned by the algorithm. Then I is a common base (see Footnote 5) with $I \subseteq E(\mathcal{C})$, where $\mathcal{C}$ is the current multichain. Since I was returned by the algorithm, we have $|I \cap C_i| = \mathrm{rank}(C_i)$ for all $C_i \in \mathcal{C}$ and this implies $\mathrm{span}(I \cap C_i) = C_i$ for all $C_i \in \mathcal{C}$ by Observation 3.1.

In order to prove that I is a popular common base, let us first prune the multichain $\mathcal{C}$ to a chain $\mathcal{C}'$, i.e., $\mathcal{C}'$ contains a single occurrence of each $C_i \in \mathcal{C}$; we will also remove any occurrence of $\emptyset$ from $\mathcal{C}'$. Observe that $E(\mathcal{C}) \subseteq E(\mathcal{C}')$: indeed, if $C_i = C_{i+1}$ in $\mathcal{C}$, then no element $e \in E$ can have $\mathrm{lev}_{\mathcal{C}}(e) = i + 1$, and hence no element gets deleted from $E(\mathcal{C})$ by pruning C_{i+1} from $\mathcal{C}$. Thus $I \subseteq E(\mathcal{C}) \subseteq E(\mathcal{C}')$. This implies that $\mathcal{C}' = \{C'_1, \ldots, C'_{p'}\}$ satisfies $\emptyset \subsetneq C'_1 \subsetneq \cdots \subsetneq C'_{p'} = E$, $I \subseteq E(\mathcal{C}')$, and $\mathrm{span}(I \cap C'_i) = C'_i$ for all $C'_i \in \mathcal{C}'$.[6] Hence I is a popular common base by Lemma 2.2.

We will now show that the algorithm always returns a popular common base, if the input instance admits one. Let B be any popular common base of M_{part} and M and let $\mathcal{D} = \{D_1, \ldots, D_q\}$ be a dual certificate for B.

Claim 3.2. *We have $q \leq |V|$ where $|\mathcal{D}| = q$.*

Proof. From the definition of dual certificate, we have $\emptyset \subsetneq D_1 \subsetneq \cdots \subsetneq D_q = E$ and $\mathrm{span}(D_i) = D_i$ for each D_i. This implies $0 < \mathrm{rank}(D_1) < \cdots < \mathrm{rank}(D_q)$. Since $\mathrm{rank}(D_q) = \mathrm{rank}(E) = |V|$, we obtain $q \leq |V|$. □

The following crucial lemma shows an invariant of the algorithm that holds for the multichain $\mathcal{C} = \{C_1, \ldots, C_p\}$ constructed in the algorithm and a dual certificate $\mathcal{D} = \{D_1, \ldots, D_q\}$ of any popular common base B. The proof will be given in this section.

Lemma 3.3. *During the execution of Algorithm 1, we always have $p \leq q$ and $D_i \subseteq C_i$ for $i = 1, \ldots, p$.*

If $p = |V| + 1$ occurs in Algorithm 1, then Lemma 3.3 implies $q \geq |V| + 1$. This contradicts Claim 3.2. Hence it has to be the case that the input instance has no popular common base when $p = |V| + 1$. Thus, assuming Lemma 3.3, the correctness of Algorithm 1 follows.

Before we prove Lemma 3.3, we need the following claim on $E(\mathcal{C})$ and $E(\mathcal{D})$.

Claim 3.4. *Assume $p \leq q$ and $D_i \subseteq C_i$ for $i = 1, \ldots, p$. For each $e \in E$, if $\mathrm{lev}_{\mathcal{C}}(e) = \mathrm{lev}_{\mathcal{D}}(e)$ and $e \in E(\mathcal{D})$, then $e \in E(\mathcal{C})$.*

[6] In fact, it will turn out that $\mathcal{C} = \mathcal{C}'$, i.e., the final $\mathcal{C}$ obtained by the algorithm itself is a dual certificate of I if the algorithm returns a common base I. This fact follows from Lemma 3.3 (with $\mathcal{C}'$ substituted for $\mathcal{D}$).

Proof. Suppose for the sake of contradiction that e fulfills the conditions of the claim, but $e \notin E(C)$. Let $e \in E_v$. It follows from the definition of $E(C)$ that there exists an element $e' \in E_v$ such that one of the following three conditions holds: (a) $\mathsf{lev}_C(e') \geq \mathsf{lev}_C(e)+2$, (b) $\mathsf{lev}_C(e') = \mathsf{lev}_C(e)+1$ and $e \not\succ_v e'$, or (c) $\mathsf{lev}_C(e') = \mathsf{lev}_C(e)$ and $e' \succ_v e$.

Because $D_i \subseteq C_i$ for each $i \in \{1, \dots, p\}$, we have $\mathsf{lev}_{\mathcal{D}}(e') \geq \mathsf{lev}_C(e')$. Since $\mathsf{lev}_{\mathcal{D}}(e) = \mathsf{lev}_C(e)$, the existence of such an $e' \in E_v$ implies $e \notin E(\mathcal{D})$, a contradiction. Thus we have $e \in E(C)$. □

The proof of Lemma 3.3 will use the following fact, known as the strong exchange property, that is satisfied by any matroid.[7]

Fact 3.5 (Brualdi [4]). *For any $X, Y \in \mathcal{I}$ and $e \in X \setminus Y$, if $Y + e \notin \mathcal{I}$, then there exists an element $f \in Y \setminus X$ such that $X - e + f$ and $Y + e - f$ are in $\mathcal{I}$.*

Now we provide the proof of Lemma 3.3. As mentioned above, this completes the proof of the correctness of our algorithm, and hence we can conclude Theorem 1.5.

Proof of Lemma 3.3. Algorithm 1 starts with $C = \{E\}$. Then the conditions in Lemma 3.3 hold at the beginning. We show by induction that they are preserved through the algorithm.

It is easy to see that the condition $p \leq q$ is preserved. Indeed, whenever Algorithm 1 is going to increase p (in line 8), it is the case that $p + 1 \leq q$ because $D_p \subseteq C_p \subsetneq E = D_q$ by the induction hypothesis. Thus $p \leq q$ is maintained in the algorithm.

We now show that $D_i \subseteq C_i$ $(i = 1, \dots, p)$ is maintained. Note that C is updated in lines 7 or 8. The update in line 8 (adding $C_p = E$) clearly preserves the condition. We complete the proof by showing that the update in line 7 also preserves the condition, i.e., we show the following statement.

- Let $C = \{C_1, \dots, C_p\}$ be a multichain with $C_p = E$ such that $p \leq q$ and $D_i \subseteq C_i$ for $i = 1, \dots, p$. Suppose that the following two conditions hold.
  (1) I is a lex-maximal common independent set subject to $I \subseteq E(C)$.
  (2) $\mathrm{span}(I \cap C_i) = C_i$ for $i = 1, \dots, k-1$, and $\mathrm{span}(I \cap C_k) \subsetneq C_k$.
  Then $D_k \subseteq \mathrm{span}(I \cap C_k)$.

To show this statement, assume for contradiction that $D_k \not\subseteq \mathrm{span}(I \cap C_k)$.

We will first show the existence of distinct elements e_1 and f_1 such that $e_1, f_1 \in E_{v_1}$ for some $v_1 \in V$ and $f_1 \in B \setminus I$ while $e_1 \in I \setminus B$. Then we will use the pair e_1, f_1 to show the existence of another pair e_2, f_2 such that $e_2, f_2 \in E_{v_2}$ where $f_2 \neq f_1$ and $f_2 \in B \setminus I$ while $e_2 \in I \setminus B$. In this manner, for any $t \in \mathbb{Z}_+$ we will be able to show *distinct* elements $f_1, f_2, \dots, f_t$ that belong to B. However, B has only $|V|$ elements, a contradiction. Then we can conclude that our assumption $D_k \not\subseteq \mathrm{span}(I \cap C_k)$ is wrong. The following is our starting claim.

Claim 3.6. *There exists $v_1 \in V$ such that there are $e_1, f_1 \in E_{v_1}$ satisfying the following properties:*
(1) $f_1 \in B \setminus I$, $I_1 := (I \cap C_k) + f_1 \in \mathcal{I}$, $I_1 \subseteq E(C)$, and $\mathsf{lev}_C(f_1) = k$,
(2) $e_1 \in I_1 \setminus B$ and $\mathsf{lev}_C(e_1) = \mathsf{lev}_{\mathcal{D}}(e_1) \leq k$.

Claim proof. Since $\mathcal{D}$ is a dual certificate of B, we have $\mathrm{span}(B \cap D_k) = D_k$. So $D_k \not\subseteq \mathrm{span}(I \cap C_k)$ implies that $\mathrm{span}(B \cap D_k) \not\subseteq \mathrm{span}(I \cap C_k)$. Hence $B \cap D_k \not\subseteq \mathrm{span}(I \cap C_k)$. So there exists f_1 such that $f_1 \in B \cap D_k$ and $f_1 \notin \mathrm{span}(I \cap C_k)$.

Since $D_k \subseteq C_k$, we have $f_1 \in D_k \subseteq C_k$. We also have $D_{k-1} \subseteq C_{k-1} = \mathrm{span}(I \cap C_{k-1}) \subseteq \mathrm{span}(I \cap C_k) \not\ni f_1$. Hence $f_1 \in C_k \setminus C_{k-1}$ and $f_1 \in D_k \setminus D_{k-1}$, i.e., $\mathsf{lev}_C(f_1) = \mathsf{lev}_{\mathcal{D}}(f_1) = k$.

[7] The original statement in [4] claims this property only for pairs of bases (maximal independent sets), but it is equivalent to Fact 3.5. Indeed, if we consider the $\mathrm{rank}(E)$-truncation of the direct sum of $(E, \mathcal{I})$ and a free matroid whose rank is $\mathrm{rank}(E)$, then the axiom in [4] applied to this new matroid implies Fact 3.5 for $(E, \mathcal{I})$.

Since $f_1 \in B \subseteq E(\mathcal{D})$ and $\mathsf{lev}_{C}(f_1) = \mathsf{lev}_{\mathcal{D}}(f_1)$, we have $f_1 \in E(C)$ by Claim 3.4. As $I \subseteq E(C)$, we then have $I_1 := (I \cap C_k) + f_1 \subseteq E(C)$. Also, $I_1 \in \mathcal{I}$ by $f_1 \notin \mathrm{span}(I \cap C_k)$. Since $\mathsf{lev}_C(f_1) = k$, the set $I_1 = (I \cap C_k) + f_1$ is lexicographically better than I. Then, the lex-maximality of I implies that I_1 must be dependent in M_{part}, i.e., there exists $e_1 \in I_1$ such that $e_1 \neq f_1$ and $e_1, f_1 \in E_{v_1}$ for some $v_1 \in V$.

We have $\mathsf{lev}_C(e_1) \leq k$ as $e_1 \in I_1 - f_1 = I \cap C_k$. Since $f_1 \in E_{v_1} \cap B$ and $|E_{v_1} \cap B| \leq 1$, we have $e_1 \notin B$. Note that $f_1 \in E(\mathcal{D})$ implies $\mathsf{lev}_{\mathcal{D}}(f_1) \geq \mathsf{lev}_{\mathcal{D}}(e_1) - 1$ and $e_1 \in E(C)$ implies $\mathsf{lev}_C(e_1) \geq \mathsf{lev}_C(f_1) - 1$. Note also that, for any element $e \in E$, we have $\mathsf{lev}_{\mathcal{D}}(e) \geq \mathsf{lev}_C(e)$ because $D_i \subseteq C_i$ for all i.

- If $f_1 \succ_{v_1} e_1$, then $\mathsf{lev}_C(e_1) > \mathsf{lev}_C(f_1)$ by $e_1 \in E(C)$,[8] and hence $\mathsf{lev}_{\mathcal{D}}(f_1) \geq \mathsf{lev}_{\mathcal{D}}(e_1) - 1 \geq \mathsf{lev}_C(e_1) - 1 \geq \mathsf{lev}_C(f_1)$. As we have $\mathsf{lev}_{\mathcal{D}}(f_1) = \mathsf{lev}_C(f_1)$, all the equalities hold.
- If $e_1 \succ_{v_1} f_1$, then $\mathsf{lev}_{\mathcal{D}}(f_1) > \mathsf{lev}_{\mathcal{D}}(e_1)$ by $f_1 \in E(\mathcal{D})$, and hence $\mathsf{lev}_{\mathcal{D}}(f_1) \geq \mathsf{lev}_{\mathcal{D}}(e_1) + 1 \geq \mathsf{lev}_C(e_1) + 1 \geq \mathsf{lev}_C(f_1)$. As we have $\mathsf{lev}_{\mathcal{D}}(f_1) = \mathsf{lev}_C(f_1)$, all the equalities hold.
- If $f_1 \sim_{v_1} e_1$, then $\mathsf{lev}_C(e_1) \geq \mathsf{lev}_C(f_1)$ by $e_1 \in E(C)$; also $\mathsf{lev}_{\mathcal{D}}(f_1) \geq \mathsf{lev}_{\mathcal{D}}(e_1)$ by $f_1 \in E(\mathcal{D})$. Hence, we have $\mathsf{lev}_{\mathcal{D}}(f_1) \geq \mathsf{lev}_{\mathcal{D}}(e_1) \geq \mathsf{lev}_C(e_1) \geq \mathsf{lev}_C(f_1)$. Since $\mathsf{lev}_{\mathcal{D}}(f_1) = \mathsf{lev}_C(f_1)$, all the equalities hold.

Thus in all the cases, we have $\mathsf{lev}_C(e_1) = \mathsf{lev}_{\mathcal{D}}(e_1) \leq k$ and $e_1 \in I_1 \setminus B$. ◁

Our next claim is the following. Recall that $I_1 := (I \cap C_k) + f_1 \in \mathcal{I}$.

Claim 3.7. *There exists $v_2 \in V$ such that there are $e_2, f_2 \in E_{v_2}$ satisfying the following properties:*
(1) $f_2 \in B \setminus I_1$, $I_2 := I_1 - e_1 + f_2 \in \mathcal{I}$, $I_2 \subseteq E(C)$, and $\mathsf{lev}_C(e_1) = \mathsf{lev}_C(f_2)$,
(2) $e_2 \in I_2 \setminus B$ and $\mathsf{lev}_C(e_2) = \mathsf{lev}_{\mathcal{D}}(e_2) \leq k$.

Claim proof. We know from Claim 3.6 that $I_1 = (I \cap C_k) + f_1 \in \mathcal{I}$. Observe that the set I_1 satisfies $\mathrm{span}(I_1 \cap C_i) = \mathrm{span}(I \cap C_i) = C_i$ for each $1 \leq i \leq k-1$; this is because $I_1 \cap C_i = I \cap C_i$ for each $i \leq k-1$. Let us apply the exchange axiom in Fact 3.5 to $I_1, B \in \mathcal{I}$ and $e_1 \in I_1 \setminus B$. Since B is maximal in $\mathcal{I}$, we have $B + e_1 \notin \mathcal{I}$, and hence there exists $f_2 \in B \setminus I_1$ such that $I_1 - e_1 + f_2$ and $B + e_1 - f_2$ are in $\mathcal{I}$.

Using that $\mathrm{span}(B \cap D_i) = D_i$ for $1 \leq i \leq q$, from $e_1 \notin \mathrm{span}(B - f_2)$ we obtain $\mathsf{lev}_{\mathcal{D}}(f_2) \leq \mathsf{lev}_{\mathcal{D}}(e_1)$: indeed, assuming $\mathsf{lev}_{\mathcal{D}}(f_2) = \ell \geq 2$ we get $D_{\ell-1} = \mathrm{span}(B \cap D_{\ell-1}) \subseteq \mathrm{span}(B - f_2)$, which implies $e_1 \notin D_{\ell-1}$ and hence also $\mathsf{lev}_{\mathcal{D}}(e_1) \geq \ell = \mathsf{lev}_{\mathcal{D}}(f_2)$. Similarly, from $f_2 \notin \mathrm{span}(I_1 - e_1)$, $\mathsf{lev}_C(e_1) \leq k$, and $\mathrm{span}(I_1 \cap C_i) = C_i$ for $1 \leq i \leq k-1$, we obtain $\mathsf{lev}_C(e_1) \leq \mathsf{lev}_C(f_2)$. Thus we have $\mathsf{lev}_C(e_1) \leq \mathsf{lev}_C(f_2) \leq \mathsf{lev}_{\mathcal{D}}(f_2) \leq \mathsf{lev}_{\mathcal{D}}(e_1) = \mathsf{lev}_C(e_1)$, implying all the equalities. Hence we have

$$f_2 \in B \setminus I_1, \quad \mathsf{lev}_C(f_2) = \mathsf{lev}_{\mathcal{D}}(f_2), \quad \mathsf{lev}_C(e_1) = \mathsf{lev}_C(f_2).$$

As $f_2 \in B \subseteq E(\mathcal{D})$, Claim 3.4 implies $f_2 \in E(C)$.

Observe that $I_2 := I_1 - e_1 + f_2 = (I \cap C_k) + f_1 - e_1 + f_2 \subseteq E(C)$, and recall $I_2 \in \mathcal{I}$. Since $\mathsf{lev}_C(e_1) = \mathsf{lev}_C(f_2)$ and $\mathsf{lev}_C(f_1) = k$, I_2 is lexicographically better than I. This implies that I_2 must be dependent in M_{part}. By the same argument as used in Claim 3.6 to show $\mathsf{lev}_C(e_1) = \mathsf{lev}_{\mathcal{D}}(e_1)$, we see that there exists e_2 such that $e_2, f_2 \in E_{v_2}$ for some $v_2 \in V$, satisfying

$$e_2 \in I_2 \setminus B, \quad \mathsf{lev}_C(e_2) = \mathsf{lev}_{\mathcal{D}}(e_2) \leq k.$$

This completes the proof of this claim. ◁

Note that $f_2 \neq f_1$ since $f_1 \in I_1$ and $f_2 \in B \setminus I_1$. Let $t \in \mathbb{Z}_+$. As shown in Claim 3.7 for $t = 3$, suppose we have constructed for $2 \leq j \leq t-1$:

[8]Actually, the case $f_1 \succ_{v_1} e_1$ is impossible because $\mathsf{lev}_C(e_1) > \mathsf{lev}_C(f_1)$ contradicts $\mathsf{lev}_C(e_1) \leq k = \mathsf{lev}_C(f_1)$. We write the proof in this form because the proofs of Claims 3.7 and 3.8 refer to the argument here to apply it to e_j, f_j, where $\mathsf{lev}_C(f_j) = k$ is not assumed.

(1) $f_j \in B \setminus I_{j-1},\ I_j := I_{j-1} - e_{j-1} + f_j \in \mathcal{I},\ I_j \subseteq E(C),$ and $\mathrm{lev}_C(e_{j-1}) = \mathrm{lev}_C(f_j)$,
(2) $e_j \in I_j \setminus B$ and $\mathrm{lev}_C(e_j) = \mathrm{lev}_{\mathcal{D}}(e_j) \leq k$.

For each j with $2 \leq j \leq t-1$, note that I_j satisfies $\mathrm{span}(I_j \cap C_i) = \mathrm{span}(I \cap C_i) = C_i$ for each i with $1 \leq i \leq k-1$. Indeed, since $\mathrm{lev}_C(e_{j-1}) = \mathrm{lev}_C(f_j)$, we have $|I_j \cap C_i| = |I \cap C_i| = \mathrm{rank}(C_i)$ for each i with $1 \leq i \leq k-1$. This implies $\mathrm{span}(I_j \cap C_i) = C_i$. Claim 3.8 generalizes Claim 3.7 for any $t \geq 3$.

Claim 3.8. *There exists $v_t \in V$ such that there are $e_t, f_t \in E_{v_t}$ satisfying the following properties:*
(1) $f_t \in B \setminus I_{t-1},\ I_t := I_{t-1} - e_{t-1} + f_t \in \mathcal{I}, I_t \subseteq E(C)$, and $\mathrm{lev}_C(e_{t-1}) = \mathrm{lev}_C(f_t)$,
(2) $e_t \in I_t \setminus B$ and $\mathrm{lev}_C(e_t) = \mathrm{lev}_{\mathcal{D}}(e_t) \leq k$.

Claim proof. Let us apply the exchange axiom in Fact 3.5 to $I_{t-1}, B \in \mathcal{I}$ and $e_{t-1} \in I_{t-1} \setminus B$. Since $B + e_{t-1} \notin \mathcal{I}$, there exists $f_t \in B \setminus I_{t-1}$ such that $I_{t-1} - e_{t-1} + f_t$ and $B + e_{t-1} - f_t$ are in $\mathcal{I}$.

By the conditions $\mathrm{span}(B \cap D_i) = D_i$ for $1 \leq i \leq q$ we have $\mathrm{lev}_{\mathcal{D}}(f_t) \leq \mathrm{lev}_{\mathcal{D}}(e_{t-1})$, and by $\mathrm{span}(I_{t-1} \cap C_i) = C_i$ for $1 \leq i \leq k-1$ and $\mathrm{lev}_C(e_{t-1}) \leq k$ we have $\mathrm{lev}_C(e_{t-1}) \leq \mathrm{lev}_C(f_t)$. Then $\mathrm{lev}_C(e_{t-1}) \leq \mathrm{lev}_C(f_t) \leq \mathrm{lev}_{\mathcal{D}}(f_t) \leq \mathrm{lev}_{\mathcal{D}}(e_{t-1}) = \mathrm{lev}_C(e_{t-1})$, and hence all the equalities hold.

So we have $f_t \in B \setminus I_{t-1}, \mathrm{lev}_C(f_t) = \mathrm{lev}_{\mathcal{D}}(f_t)$, and $\mathrm{lev}_C(e_{t-1}) = \mathrm{lev}_C(f_t)$. As $f_t \in B \subseteq E(\mathcal{D})$, Claim 3.4 implies $f_t \in E(C)$.

Observe that $I_t := I_{t-1} - e_{t-1} + f_t = (I \cap C_k) + f_1 - e_1 + \ldots + f_{t-1} - e_{t-1} + f_t \subseteq E(C)$, and recall $I_t \in \mathcal{I}$. Since $\mathrm{lev}_C(e_{j-1}) = \mathrm{lev}_C(f_j)$ for $2 \leq j \leq t$ and $\mathrm{lev}_C(f_1) = k$, the set I_t is lexicographically better than I. This implies that I_t must be dependent in M_{part}. By the same argument as used in Claim 3.6 to show $\mathrm{lev}_C(e_1) = \mathrm{lev}_{\mathcal{D}}(e_1)$, we see that there exists e_t such that $e_t, f_t \in E_{v_t}$ for some v_t, satisfying also $e_t \in I_t \setminus B$ and $\mathrm{lev}_C(e_t) = \mathrm{lev}_{\mathcal{D}}(e_t) \leq k$. This completes the proof of this claim. $\lhd$

Observe that f_t is distinct from $f_1, \ldots, f_{t-1}$ since $\{f_1, \ldots, f_{t-1}\} \subseteq I_{t-1}$ while $f_t \in B \setminus I_{t-1}$. Thus, for each $t \in \mathbb{Z}_+$, we have shown distinct elements $f_1, \ldots, f_t$ in B, contradicting that $|B| \leq |V|$. Therefore, it has to be the case that $D_k \subseteq \mathrm{span}(I \cap C_k)$.

This completes the proof of Lemma 3.3. □

4 POPULAR COMMON INDEPENDENT SET POLYTOPE

This section proves Corollary 1.6 and Theorem 1.7.

As a motivating example, we start by describing the popular colorful forest problem. Let $G = (U, E)$ be an undirected graph where $E = E_1 \cup \cdots \cup E_n$, i.e., E is partitioned into n color classes. Equivalently, there are n agents $1, \ldots, n$ where agent i owns the elements in E_i. For each i, there is a partial order $\succ_i$ over elements in E_i.

Recall that $S \subseteq E$ is a *colorful forest* if (i) S is a forest in G and (ii) $|S \cap E_i| \leq 1$ for every $i \in \{1, \ldots, n\}$. That is, we seek an acyclic network with diversity, i.e., there is at most one edge from each color class. We refer to Section 1 on how every agent compares any pair of colorful forests; for any pair of colorful forests F and F', let $\phi(F, F')$ be the number of agents that prefer F to F'. A colorful forest F is *popular* if $\phi(F, F') \geq \phi(F', F)$ for any colorful forest F'. The popular colorful forest problem is to decide if a given instance G admits a popular colorful forest or not. Observe that a colorful forest is a common independent set in the intersection of the partition matroid defined by $E_1 \cup \cdots \cup E_n$ and the graphic matroid of G. Therefore, this problem is a special case of the popular common independent set problem.

An input of the general popular common independent set problem is essentially the same as for the popular common base problem: we are given a partition matroid M_{part} over a set E with the partition classes of M_{part} being indexed by a finite set V, and with each $v \in V$ having a partial order $\succ_v$ over the partition class E_v; we are also given an arbitrary matroid $M = (E, \mathcal{I})$. Here, it is not necessarily assumed that the rank of M is $|V|$. We say that $v \in V$ prefers a common independent set I to a common independent set I' if either (i) $I \cap E_v$ is a singleton while $I' \cap E_v$ is empty, or (ii) both $I \cap E_v$ and $I' \cap E_v$ are singletons and v prefers the element in $I \cap E_v$ to that in $I' \cap E_v$. The value $\phi(I, I')$ denotes the number of indices in V that prefer I to I'.

Definition 4.1. A common independent set I of the matroids M_{part} and M is popular if $\phi(I, I') \geq \phi(I', I)$ for all common independent sets I'.

We now show that Algorithm 1 solves the popular common independent set problem. To this end, we will construct an auxiliary instance whose common bases correspond to common independent sets of the original instance.

An auxiliary instance. For each $v \in V$, add a dummy element e_v to E and call the resulting ground set $\hat{E}$. The new partition matroid $\hat{M}_{\text{part}}$ is defined by the partition $\hat{E} = \dot\bigcup_{v \in V}(E_v \cup \{e_v\})$. Furthermore, for each $v \in V$, the dummy element e_v will be the *worst* element in v's preference order $\succ_v$, that is, every $f \in E_v$ satisfies $f \succ_v e_v$. To define the other new matroid $\hat{M}$, first we add the dummy elements $\{ e_v : v \in V \}$ to the ground set E of M as *free* elements, i.e., for any $v \in V$, any set $S \subseteq \hat{E}$ excluding e_v cannot span e_v. As we want to reduce the current problem to the popular common base problem, we further truncate the matroid with size $|V|$, i.e., all sets of size larger than $|V|$ will be deleted from the independent set family. Let $\hat{M}$ denote the resultant matroid.

Observe that there exists a one-to-one correspondence between common independent sets of M_{part} and M and common bases of $\hat{M}_{\text{part}}$ and $\hat{M}$. Suppose I is a common independent set of M_{part} and M and let $W \subseteq V$ be the set of indices $v \in V$ with $I \cap E_v = \emptyset$. Let $B = I \cup \bigcup_{v \in W}\{e_v\}$. Then B is a common base of $\hat{M}_{\text{part}}$ and $\hat{M}$. Conversely, given a common base B of $\hat{M}_{\text{part}}$ and $\hat{M}$, we can obtain a common independent set I of M_{part} and M by deleting the dummy elements.

Popularity in the original and auxiliary instances. Let I and I' be common independent sets of the original matroids M_{part} and M and let B and B' be the common bases of the auxiliary matroids $\hat{M}_{\text{part}}$ and $\hat{M}$ that correspond to I and I', respectively. Observe that $\phi(I, I') = \phi(B, B')$. Thus, popular common independent sets of M_{part} and M correspond to popular common bases of $\hat{M}_{\text{part}}$ and $\hat{M}$, and vice versa. Therefore, the popular common independent set problem reduces to the popular common base problem, and hence can be solved by Algorithm 1. From this, Corollary 1.6 follows.

The popular common independent set polytope. Every popular common base B of the auxiliary matroids $\hat{M}_{\text{part}}$ and $\hat{M}$ has a dual certificate as given in Lemma 2.1[9] and Lemma 2.2. We will now show that dual certificates of the auxiliary instance are even more special than what is given in Lemma 2.2—along with the properties described there, the following property is also satisfied.

Lemma 4.2. *Let B be a popular common base of the auxiliary matroids $\hat{M}_{\text{part}}$ and $\hat{M}$ and let $C = \{C_1, \dots, C_p\}$ be a dual certificate for B. Then $p \leq 2$.*

Proof. Suppose not, i.e., $p \geq 3$. From the definition of a dual certificate C, we have $\emptyset \subsetneq C_1 \subsetneq C_2 \subsetneq \cdots \subsetneq C_p = \hat{E}$ (see Lemma 2.2). We will now show that $B \cap C_1 = \emptyset$. Since we have $\text{span}(B \cap C_1) = C_1$, this means $C_1 = \emptyset$; however this contradicts $C_1 \neq \emptyset$. This will give us the desired contradiction, proving $p \leq 2$.

In order to show that $B \cap C_1 = \emptyset$, it suffices to prove that for each $v \in V$, the unique element in $B \cap (E_v \cup \{e_v\})$, denoted by $B(v)$, is not contained in C_1.

- If $B(v) \neq e_v$, then the dummy element e_v is not in B. Since e_v is not spanned by any set $S \subseteq \hat{E}$ with $e_v \notin S$ and $\text{rank}(S) < |V|$, the condition $\text{span}(B \cap C_j) = C_j$, yielding also $|B \cap C_j| = \text{rank}(C_j)$, for all $j = 1, 2, \dots, p$ implies that $e_v \notin C_j$ for any $j < p$. Hence

[9] In LP1 and LP2 defined with respect to B, the set E_v will be replaced by $E_v \cup \{e_v\}$ and the rank function of $\hat{M}$ will be used.

$\mathsf{lev}_{\mathcal{C}}(e_v) = p$, which implies that every edge in $E(\mathcal{C}) \cap E_v$ has level either p or $p-1$. Because $p \geq 3$, this means that no element of C_1 is present in $E(\mathcal{C}) \cap E_v$. Thus we have $B(v) \notin C_1$.

- If $B(v) = e_v$, then $e_v \in E(\mathcal{C})$. This implies $\mathsf{lev}_{\mathcal{C}}(e_v) > 1$ because e_v is the worst element in $E_v \cup \{e_v\}$. Hence $B(v)$ is not in C_1.

In both cases, $B(v) \notin C_1$ for every $v \in V$. Thus we have $B \cap C_1 = \emptyset$, as desired. □

Lemma 4.2 shows that any dual certificate $\mathcal{C}$ for a popular common base B in the auxiliary instance has length at most 2, i.e., B has a dual certificate either of the form $\mathcal{C} = \{\hat{E}\}$ or of the form $\mathcal{C} = \{C, \hat{E}\}$. Let B be the popular common base computed by Algorithm 1 in the auxiliary instance, and let $\mathcal{C}$ be a dual certificate for B. The following lemma shows that if preferences are weak rankings, then $\mathcal{C}$ is a dual certificate for all popular common bases. Note that this proof crucially uses the fact that preferences are weak rankings—recall that we use this assumption in Theorem 1.7 as well. Indeed, assuming weak rankings is indispensable there, since the min-cost popular common independent set problem for partial order preferences is NP-hard, due to the NP-hardness of its special case, the min-cost popular branching problem with partial order preferences [22].

Lemma 4.3. *Assume that preferences are weak rankings and suppose that B is the popular common base computed by Algorithm 1 applied to the auxiliary instance $\hat{M}_{\rm part}$ and $\hat{M}$, and that $\mathcal{C}$ is a dual certificate for B. Then for any arbitrary popular common base B' in the auxiliary instance, we have (i) $B' \subseteq E(\mathcal{C})$ and (ii) if $\mathcal{C} = \{C, \hat{E}\}$, then $|B' \cap C| = \mathrm{rank}(C)$.*

Proof. Let $(\vec{y}, \vec{\alpha})$ be the dual variables defined from $\mathcal{C}$ as given in Lemma 2.1. That is, $y_{\hat{C}} = 1$ for each $\hat{C} \in \mathcal{C}$ and $y_S = 0$ for any other $S \subseteq \hat{E}$, and $\alpha_v = -|\{\,\hat{C} \in \mathcal{C} : B(v) \in \hat{C}\,\}|$ for every $v \in V$. Note that the length of $\mathcal{C}$ is at most two by Lemma 4.2.

Consider LP1 and LP2 defined with respect to B. Since both B and B' are popular, their characteristic vectors are both optimal solutions to LP1. Since $(\vec{y}, \vec{\alpha})$ is an optimal solution to LP2, if $\mathcal{C} = \{C, \hat{E}\}$ then we have $|B' \cap C| = \mathrm{rank}(C)$ by complementary slackness. It remains to show that $B' \subseteq E(\mathcal{C})$. We distinguish between two cases, depending on whether the length of $\mathcal{C}$ is one or two.

(1) Suppose that $\mathcal{C} = \{\hat{E}\}$. Let $\mathcal{D}$ be a dual certificate of B' as described in Lemma 2.2. Then $B' \subseteq E(\mathcal{D})$. Assume that $\mathcal{D} = \{D, \hat{E}\}$, as otherwise $\mathcal{D} = \{\hat{E}\} = \mathcal{C}$.
Take any $v \in V$. We are going to show that $B'(v) \in E(\mathcal{C})$. If $B'(v) \in D$ then we have $\mathsf{lev}_{\mathcal{D}}(B'(v)) = 1 = \mathsf{lev}_{\mathcal{C}}(B'(v))$; along with $B'(v) \in E(\mathcal{D})$, this implies $B'(v) \in E(\mathcal{C})$ by Claim 3.4. We thus assume that $B'(v) \notin D$.
Since the characteristic vector $\vec{x}$ of B and $\vec{x}'$ of B' are optimal solutions to LP1 (defined with respect to B) and $(\vec{y}, \vec{\alpha})$ is an optimal solution to LP2 (its dual LP), we will use complementary slackness. Because $x_{B(v)} = 1$, we have $\sum_{\hat{C} \in \mathcal{C}: B(v) \in \hat{C}} y_{\hat{C}} + \alpha_v = \mathsf{wt}_B(B(v)) = 0$. Similarly, because $x'_{B'(v)} = 1$, we have $\sum_{\hat{C} \in \mathcal{C}: B'(v) \in \hat{C}} y_{\hat{C}} + \alpha_v = \mathsf{wt}_B(B'(v))$. By subtracting the former from the latter, we obtain

$$\sum_{\hat{C} \in \mathcal{C}: B'(v) \in \hat{C}} y_{\hat{C}} \;-\; \sum_{\hat{C} \in \mathcal{C}: B(v) \in \hat{C}} y_{\hat{C}} \;=\; \mathsf{wt}_B(B'(v)). \tag{2}$$

Since $\mathcal{C} = \{\hat{E}\}$, the left-hand side is $1 - 1 = 0$. By this $\mathsf{wt}_B(B'(v)) = 0$, which implies $B(v) \sim_v B'(v)$. The fact $B(v) \in E(\mathcal{C})$ implies that $B(v)$ is maximal with respect to $\succ_v$ in $E_v \cup \{e_v\}$. Because $\succ_v$ is a weak ranking, $B(v) \sim_v B'(v)$ means that $B'(v)$ is also maximal, and hence $B'(v) \in E(\mathcal{C})$ follows.

(2) Suppose now that $\mathcal{C} = \{C, \hat{E}\}$. Let $\mathcal{D}$ be a dual certificate of B'. Then we have $\mathcal{D} = \{D, \hat{E}\}$ and $D \subseteq C$ by Lemma 3.3. Take any $v \in V$. We are going to show that $B'(v) \in E(\mathcal{C})$. If

$B'(v) \notin C$ (respectively, if $B'(v) \in D$), then $B'(v) \notin D$ (respectively, $B'(v) \in C$); hence in these cases we get $\mathsf{lev}_{\mathcal{C}}(B'(v)) = \mathsf{lev}_{\mathcal{D}}(B'(v))$. This fact along with $B'(v) \in E(\mathcal{D})$ implies that $B'(v) \in E(\mathcal{C})$, by Claim 3.4. Therefore, let us assume that $B'(v) \in C \setminus D$.
By the same analysis as given in Case 1, Equation (2) holds. Let us also consider LP1 and LP2 defined with respect to B' (instead of B). Let $(\vec{z}, \vec{\beta})$ be the optimal solution of LP2 corresponding to $\mathcal{D}$. As before, the characteristic vectors of B and B' are optimal solutions to LP1. By the same argument (with B', B and $\mathcal{D}$ taking the places of B, B', and $\mathcal{C}$, respectively), we have:

$$\sum_{\hat{D} \in \mathcal{D} : B(v) \in \hat{D}} z_{\hat{D}} - \sum_{\hat{D} \in \mathcal{D} : B'(v) \in \hat{D}} z_{\hat{D}} = \mathsf{wt}_{B'}(B(v)). \tag{3}$$

Since $B'(v) \in C$, the left-hand side of (2) is 1 or 0, and so is $\mathsf{wt}_B(B'(v))$, which implies that we have $B'(v) \succ_v B(v)$ or $B'(v) \sim_v B(v)$. Furthermore, since $B'(v) \notin D$, the left-hand side of (3) is 1 or 0, and so is $\mathsf{wt}_{B'}(B(v))$, which implies that $B(v) \succ_v B'(v)$ or $B(v) \sim_v B'(v)$. Therefore we must have $B'(v) \sim_v B(v)$. Hence $B(v) \in C$ follows from (2).
We have shown that $B'(v) \sim_v B(v)$ and $B(v) \in C$. We also have $B'(v) \in C$. Since $B(v) \in E(\mathcal{C})$, we see that $B(v)$ is maximal in $C \cap (E_v \cup \{e_v\})$ and dominates all elements in $(E_v \cup \{e_v\}) \setminus C$ with respect to $\succ_v$. Since $\succ_v$ is a weak ranking and $B'(v) \sim_v B(v)$, the element $B'(v) \in C$ also satisfies these conditions, and hence $B'(v) \in E(\mathcal{C})$.

Thus, in both cases we have $B'(v) \in E(\mathcal{C})$ for every $v \in V$. Hence $B' \subseteq E(\mathcal{C})$. □

By Lemma 4.3, any popular common base B' in the auxiliary instance (consisting of $\hat{M}_{\text{part}}$ and $\hat{M}$) satisfies $B' \subseteq E(\mathcal{C})$ and $|B' \cap C| = \text{rank}(C)$ if $\mathcal{C} = \{C, \hat{E}\}$. Conversely, any common base B' in the auxiliary instance that satisfies these conditions is popular by Lemma 2.2. Therefore the set of all popular common bases in the auxiliary instance can be described as a face of the matroid intersection polytope of $\hat{M}_{\text{part}}$ and $\hat{M}$. Since a popular common independent sets in the original instance (consisting of M_{part} and M) is obtained by deleting the dummy elements from popular common bases in the auxiliary instance, Theorem 1.7 follows. We now restate this result more explicitly as Theorem 4.4 below. Here, a projection means the deletion of variables corresponding to the dummy elements $\{ e_v : v \in V \}$ we have added in the construction of the auxiliary instance.

Theorem 4.4. *Assume that preferences are weak rankings and suppose that $\mathcal{C} = \{C, \hat{E}\}$ is a dual certificate for the popular common base returned by Algorithm 1 when applied to the auxiliary matroids $\hat{M}_{\text{part}}$ and $\hat{M}$. Then the popular common independent set polytope of the original instance is a projection of the polytope that is defined by the constraints $\sum_{e \in C} x_e = \text{rank}(C)$ and $x_e = 0$ for all $e \in \hat{E} \setminus E(\mathcal{C})$ along with all the constraints of LP1.*

5 POPULAR COMMON BASE WITH FORCED/FORBIDDEN ELEMENTS

We prove Theorem 1.8 in this section. Observe that the problem of deciding if there exists a popular common base B such that $B \supseteq E^+$ for a given set $E^+ \subseteq E$ of *forced* elements can be reduced to the problem of deciding if there exists a popular common base B such that certain elements are *forbidden* for B.

Let $V' \subseteq V$ be the set of those elements v such that $E_v \cap E^+ \neq \emptyset$; clearly, we may assume $|E_v \cap E^+| = 1$ for each $v \in V'$. Let $E' = \bigcup_{v \in V'} (E_v \setminus E^+)$. Since $B \supseteq E^+$ if and only if $B \cap E' = \emptyset$, it follows that the problem of deciding if there exists a popular common base B such that $E^+ \subseteq B$ and $E^- \cap B = \emptyset$ reduces to the problem of deciding if there exists a popular common base B such that $B \cap E_0 = \emptyset$ for a set $E_0 \subseteq E$ of forbidden elements.

Forbidden elements. We present our algorithm that decides if there exists a popular common base that avoids E_0 for a given subset E_0 of E as Algorithm 2. The only difference from the original popular common base algorithm (Algorithm 1) is in line 4: the new algorithm finds a lexicographically maximal common independent set in the set $E(\mathcal{C}) \setminus E_0$ instead of $E(\mathcal{C})$. Recall that $\mathrm{rank}(E) = |V|$.

Algorithm 2 The popular common base algorithm with the forbidden element set E_0

1: Initialize $p = 1$ and $C_1 = E$. ▹ Initially we set $\mathcal{C} = \{E\}$.
2: **while** $p \leq |V|$ **do**
3: Compute the set $E(\mathcal{C})$ from the current multichain $\mathcal{C} = \{C_1, \ldots, C_p\}$.
4: Find a common independent set I that lexicographically maximizes $(|I \cap C_1|, \ldots, |I \cap C_p|)$ subject to $I \subseteq E(\mathcal{C}) \setminus E_0$.
5: **if** $|I \cap C_i| = \mathrm{rank}(C_i)$ for every $i = 1, \ldots, p$ **then** return I.
6: Let k be the minimum index such that $|I \cap C_k| < \mathrm{rank}(C_k)$.
7: Update $C_k \leftarrow \mathrm{span}(I \cap C_k)$.
8: **if** $k = p$ **then** $p \leftarrow p + 1$, $C_p \leftarrow E$, and $\mathcal{C} \leftarrow \mathcal{C} \cup \{C_p\}$.
9: Return "The input instance has no popular common base that avoids E_0."

Theorem 5.1. *Let $E_0 \subseteq E$. There exists a popular common base B such that $B \cap E_0 = \emptyset$ if and only if Algorithm 2 returns a popular common base with no edge of E_0.*

Proof. The easy side is to show that if Algorithm 2 returns a common independent set I, then (i) I is popular and (ii) $I \cap E_0 = \emptyset$. As done in Section 3, let us prune the multichain $\mathcal{C}$ into a chain $\mathcal{C}'$. Because $I \subseteq E(\mathcal{C}) \setminus E_0$ and $E(\mathcal{C}) \subseteq E(\mathcal{C}')$, we have $I \subseteq E(\mathcal{C}') \setminus E_0$. Since $I \subseteq E(\mathcal{C}')$ and $|I \cap C'_i| = \mathrm{rank}(C'_i)$, and hence $\mathrm{span}(I \cap C'_i) = C'_i$, for every $C'_i \in \mathcal{C}'$, it follows from Lemma 2.2 that I is a popular common base.

We now show the converse. Suppose that there exists a popular common base B in the intersection of M_{part} and M such that $B \cap E_0 = \emptyset$. Let $\mathcal{D} = \{D_1, \ldots, D_q\}$ be a dual certificate for B. Then we have $B \subseteq E(\mathcal{D}) \setminus E_0$. It suffices to show that Algorithm 2 maintains the following invariant: the multichain $\mathcal{C} = \{C_1, \ldots, C_p\}$ maintained in the algorithm satisfies $p \leq q$ and $D_i \subseteq C_i$ for any $i = 1, 2, \ldots, p$.

We can show a variant of Lemma 3.3, i.e., we can show that when C_k is updated in the algorithm, $D_k \subseteq \mathrm{span}(I \cap C_k)$ holds where I is a lexicographically maximal common independent set in $E(\mathcal{C}) \setminus E_0$. The proof of Lemma 3.3 works almost as it is. Recall that we sequentially find elements $f_1, e_1, f_2, e_2, \ldots$ in the proof of Lemma 3.3. For each $j = 1, 2, \ldots$, in addition to the condition $f_j \in E(\mathcal{C})$, we have $f_j \notin E_0$ since $f_j \in B \subseteq E \setminus E_0$. By this, $I_j = (I \cap C_k) + f_1 - e_1 + f_2 \cdots - e_{j-1} + f_j$ satisfies $I_j \subseteq E(\mathcal{C}) \setminus E_0$ for each j. Hence the proof of Lemma 3.3 works with replacing "lex-maximality subject to $I \subseteq E(\mathcal{C})$" by "lex-maximality subject to $I \subseteq E(\mathcal{C}) \setminus E_0$". □

6 MIN-COST POPULAR ARBORESCENCE

We prove Theorem 1.9 in this section. We present a reduction from the Vertex Cover problem, whose input is an undirected graph H and an integer k, and asks whether H admits a set of k vertices that is a vertex cover, that is, contains an endpoint from each edge in H.

Our reduction is strongly based on the reduction used in [22, Theorem 6.3] which showed the NP-hardness of the min-cost popular branching problem when vertices have partial order preferences. Recall that the min-cost popular branching problem is polynomial-time solvable when vertices have weak rankings [22] (also implied by Theorem 1.7). Note also that neither the hardness of min-cost popular branching for partial order preferences [22], nor the hardness of min-cost

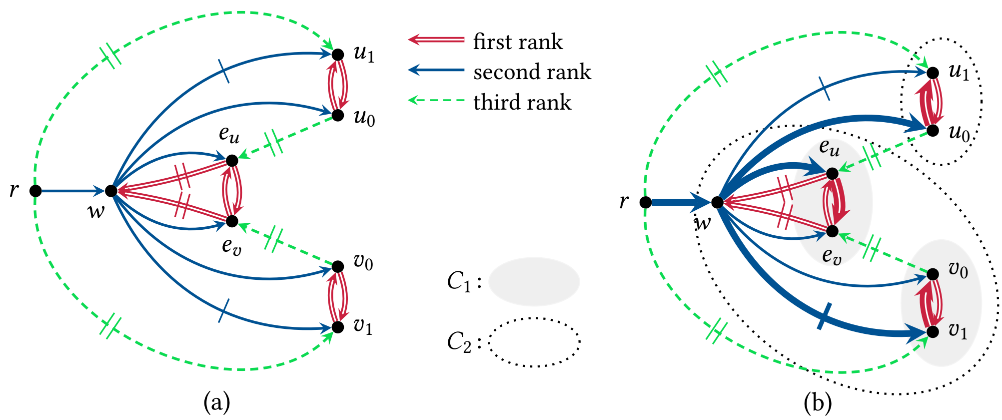


Fig. 1. Illustration of the reduction in the proof of Theorem 1.9. Figure (a) illustrates the construction showing a subgraph of G, assuming that the input graph H contains an edge $e = uv$. Edges in E_1, E_2, and E_3 are depicted with double red, single blue, and dashed green lines, respectively. Edges marked with two, one, and zero crossbars have cost ∞, 1, and 0, respectively. Figure (b) illustrates the popular arborescence A in bold, assuming $v \in S$ and $u \notin S$. The chain $C_1 \subsetneq C_2 \subsetneq C_3 = E$ certifying the popularity of A is shown using grey and dotted ellipses for edges in C_1 and C_2, respectively.

popular assignment for strict preferences [21] implies Theorem 1.9, since the min-cost popular arborescence problem with strict rankings does not contain either of these problems.

To show the NP-hardness of the min-cost popular arborescence problem when vertices have strict rankings, we construct a directed graph $G = (V \cup \{r\}, E = E_1 \cup E_2 \cup E_3)$ as follows; see Figure 1 for an illustration. We set

$$\begin{aligned} V &= \{w\} \cup \{v_0, v_1 : v \in V(H)\} \cup \{e_u, e_v : e = uv \in E(H)\}, \\ E_1 &= \{(e_u, e_v), (e_v, e_u), (e_u, w), (e_v, w) : e = uv \in E(H)\} \\ &\qquad \cup \{(v_0, v_1), (v_1, v_0) : v \in V(H)\}, \\ E_2 &= \{(r, w)\} \cup \{(w, x) : x \in V(G) \setminus \{r, w\}\}, \\ E_3 &= \{(r, v_1) : v \in V(H)\} \cup \{(u_0, e_u), (v_0, e_v) : e = uv \in E(H)\}. \end{aligned}$$

To define the preferences of each vertex in G, we let all vertices prefer edges of E_1 to edges of E_2, which in turn are preferred to edges of E_3. Whenever some vertex has more than one incoming edge in some E_i, $i \in \{1, 2, 3\}$, then it orders them in some arbitrarily fixed strict order. We set the cost of each edge in E_3, as well as the cost of all edges entering w except for (r, w) as ∞. We set the cost of (w, v_1) as 1 for each $v \in V(H)$, and we set the cost of all remaining edges as 0. We define our budget to be k, finishing the construction of our instance of min-cost popular arborescence.

We are going to show that H admits a vertex cover of size at most k if and only if G has a popular arborescence of cost at most k.

Suppose first that A is a popular arborescence in G with cost at most k. We prove that the set $S = \{v \in V(H) : (w, v_1) \in A\}$ is a vertex cover in H. Since each edge (w, v_1) has cost 1, our budget implies $|S| \leq k$.

For a vertex $v \in V(H)$ and an edge $e = uv \in E(H)$, let $A_v = A \cap (\delta(v_0) \cup \delta(v_1))$ and $A_e = A \cap (\delta(e_u) \cup \delta(e_v))$, respectively. We note that each $v \in V(H)$ satisfies that A_v is either $\{(w, v_0), (v_0, v_1)\}$

or $\{(w, v_1), (v_1, v_0)\}$. Indeed, if this is not the case, we have $A_v = \{(w, v_0), (w, v_1)\}$, since A is an arborescence with finite cost. However, this contradicts the popularity of A, since $A \setminus \{(w, v_1)\} \cup \{(v_0, v_1)\}$ is more popular than A. We can similarly show that each $e = uv \in E(H)$ satisfies that A_e is either $\{(w, e_u), (e_u, e_v)\}$ or $\{(w, e_v), (e_v, e_u)\}$. Note also that $(r, w) \in A$, as all other edges entering w have infinite cost.

Assume for the sake of contradiction that S is not a vertex cover of H, i.e., there exists an edge $e = uv \in E(H)$ such that neither (w, u_1) nor (w, v_1) is contained in A. Then we have $A_u = \{(w, u_0), (u_0, u_1)\}$ and $A_v = (w, v_0), (v_0, v_1)\}$. By symmetry, we assume without loss of generality that $A_e = \{(w, e_u), (e_u, e_v)\}$. Define an edge set A' by

$$A' = (A \setminus (A_e \cup A_v \cup \{(r, w)\})) \cup \{(r, v_1), (v_1, v_0), (v_0, e_v), (e_v, e_u), (e_u, w)\}.$$

We can see that A' is an arborescence and is more popular than A, since three vertices, v_0, e_u, and w, prefer A' to A, while two vertices, v_1 and e_v, prefer A to A', and all others are indifferent between them. This proves that S is a vertex cover of H.

For the other direction, assume that S is a vertex cover in H. We construct a popular arborescence A of cost $|S|$ in G. For each $e \in E(H)$ we fix an endpoint $\sigma(e)$ of e that is contained in S, and we denote by $\bar{\sigma}(e)$ the other endpoint of e (which may or may not be in S). Let

$$\begin{aligned} A = \{(r, w)\} &\cup \{(w, v_1), (v_1, v_0) : v \in S\} \\ &\cup \{(w, v_0), (v_0, v_1) : v \in V(H) \setminus S\} \\ &\cup \{(w, e_{\bar{\sigma}(e)}), (e_{\bar{\sigma}(e)}, e_{\sigma(e)}) : e \in E(H)\}. \end{aligned}$$

It is straightforward to verify that A is an arborescence and its cost is exactly $|S|$. Hence it remains to prove its popularity, which is done by showing a dual certificate $\mathcal{C}$ for A.

To define $\mathcal{C}$, let us first define a set $X = \{w\} \cup \{e_u, e_v : e = uv \in E(H)\} \cup \{v_0, v_1 : v \in S\}$ of vertices in G. Then we set $\mathcal{C} = \{C_1, C_2, C_3\}$ where

$$\begin{aligned} C_1 &= \{(e_u, e_v), (e_v, e_u) : e = uv \in E(H)\} \cup \{(v_0, v_1), (v_1, v_0) : v \in S\}, \\ C_2 &= \{f \in E(H) : f \text{ has two endpoints in } X\} \cup \{(v_0, v_1), (v_1, v_0) : v \in V(H) \setminus S\}, \\ C_3 &= E. \end{aligned}$$

Let us first check that $\operatorname{rank}(C_i) = |A \cap C_i|$ for each $C_i \in \mathcal{C}$. Clearly, C_1 consists of mutually vertex-disjoint 2-cycles, and A contains an edge from each of them. Thus $\operatorname{rank}(C_1) = |A \cap C_1|$ follows. The edge set C_2 consists of all edges induced by the vertices of X, together with another set of mutually vertex-disjoint 2-cycles that share no vertex with X. It is easy to verify that $A \cap C_2$ contains an edge from each of the 2-cycles in question, as well as a directed tree containing all vertices of X. Thus, $\operatorname{rank}(C_2) = |A \cap C_2|$ holds. Since A is an arborescence, $\operatorname{rank}(C_3) = \operatorname{rank}(E) = |V| = |A \cap C_3|$ is obvious. Observe that for each $i \in \{1, 2, 3\}$ we have $\operatorname{span}(C_i) = C_i$, and hence $\operatorname{rank}(C_i) = |A \cap C_i|$ implies $\operatorname{span}(A \cap C_i) = C_i$.

It remains to see that $A \subseteq E(\mathcal{C})$. First, $A(w) = (r, w)$ is the unique incoming edge of w with $\mathcal{C}$-level 3. For each $v \in S$, $\operatorname{lev}^*_{\mathcal{C}}(v_0) = 2$ while $\operatorname{lev}^*_{\mathcal{C}}(v_1) = 3$, and by their preferences both $A(v_0) = (v_1, v_0)$ and $A(v_1) = (w, v_1)$ are in $E(\mathcal{C})$. For each $v \in V(H) \setminus S$, $\operatorname{lev}^*_{\mathcal{C}}(v_0) = \operatorname{lev}^*_{\mathcal{C}}(v_1) = 3$, and hence both $A(v_0) = (w, v_0)$ and $A(v_1) = (v_0, v_1)$ are in $E(\mathcal{C})$. Finally, consider an edge $e = uv \in E(H)$ with $\sigma(e) = v \in S$. As $\operatorname{lev}^*_{\mathcal{C}}(e_u) \leq 3$, and since e_u prefers (w, e_u) to (u_0, e_u), we know that the edge $A(e_u) = (w, e_u) \in C_2$ is contained in $E(\mathcal{C})$. By contrast, since $v \in S$ implies $v_0 \in X$, we obtain $\operatorname{lev}^*_{\mathcal{C}}(e_v) = 2$, and therefore the edge $A(e_v) = (e_u, e_v) \in C_1$ is contained in $E(\mathcal{C})$. By Lemma 2.2, this proves that A is indeed a popular arborescence.

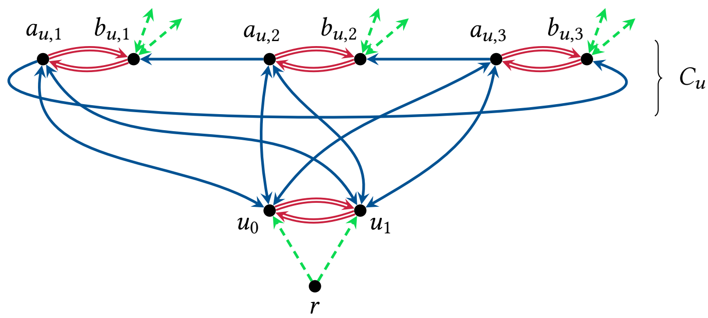


Fig. 2. Illustration of a gadget G_u in the proof of Theorem 1.10. Preferences are encoded using line types and colors as in Figure 1.

7 MINIMUM UNPOPULARITY MARGIN ARBORESCENCE

We prove Theorem 1.10 in this section. It is easy to see that the problem is in NP, since given an arborescence A we can verify $\mu(A) \leq k$ efficiently, assuming that a dual certificate for A (i.e., a solution for LP2 with objective value k) is provided.

To prove NP-hardness, we present a reduction from the following NP-hard variant of the Exact 3-Cover problem [17]. The input contains a set U of size $3n$ and a set family $\mathcal{S} = \{S_1, \ldots, S_{3n}\}$ where $S_i \subseteq U$ and $|S_i| = 3$ for each $S_i \in \mathcal{S}$, and each $u \in U$ is contained in exactly three sets from $\mathcal{S}$. The task is to decide whether there exist n sets in $\mathcal{S}$ whose union is U.

Our reduction draws inspiration from the reduction used in [22, Theorem 4.6] which proved the NP-hardness of the k-unpopularity margin branching problem when vertices have partial order preferences. Recall that this problem was shown to be polynomial-time solvable when vertices have weak rankings [22]. Note also that Theorem 1.10 does not follow from the NP-hardness of either the k-unpopularity margin branching problem [22] or the k-unpopularity margin assignment problem [21].

To show the NP-hardness of the k-unpopularity margin arborescence problem when vertices have strict rankings, we construct a directed graph $G = (V \cup \{r\}, E = E_1 \cup E_2 \cup E_3)$ as follows; see Figure 2 for an illustration. For each $u \in U$ we construct a gadget G_u whose vertex set is $\{u_0, u_1\} \cup A_u \cup B_u$ where $A_u = \{a_{u,1}, a_{u,2}, a_{u,3}\}$ and $B_u = \{b_{u,1}, b_{u,2}, b_{u,3}\}$. First we add four 2-cycles, with all their edges in E_1, on vertex sets $\{a_{u,i}, b_{u,i}\}$ for each $i = 1, 2, 3$, as well as on $\{u_0, u_1\}$; these $8|U|$ edges comprise E_1. We next add edges of E_2: first, we stitch together the three 2-cycles on $A_u \cup B_u$ with edges $(a_{u,3}, b_{u,2})$, $(a_{u,2}, b_{u,1})$, and $(a_{u,1}, b_{u,3})$; second, we add all possible edges between $\{u_0, u_1\}$ and A_u, creating a bidirected $K_{2,3}$. We denote the unique 6-cycle on $A_u \cup B_u$ as C_u. This finishes the construction of our gadget G_u. To complete the definition of G, it remains to define E_3. To this end, for each $u \in U$ we fix an arbitrary ordering over the three sets of $\mathcal{S}$ containing u, and denote them as $S(u, 1)$, $S(u, 2)$, and $S(u, 3)$. We then let

$$E_3 = \{(r, u_0), (r, u_1) : u \in U\} \cup \{(b_{u,i}, b_{v,j}) : \exists S \in \mathcal{S} \text{ such that } S = S(u, i) = S(v, j)\}.$$

To define the preferences of each vertex in G, we let all vertices prefer edges of E_1 to those in E_2, which in turn are preferred to edges of E_3. Whenever some vertex has more than one incoming edge in some E_i, $i \in \{1, 2, 3\}$, then it orders them in some fixed strict order with the only constraint that edges from u_0 are preferred to edges from u_1 for each $u \in U$.

We are going to show that our instance of Exact 3-Cover is solvable if and only if G admits an arborescence with $\mu(A) \leq 2n$.

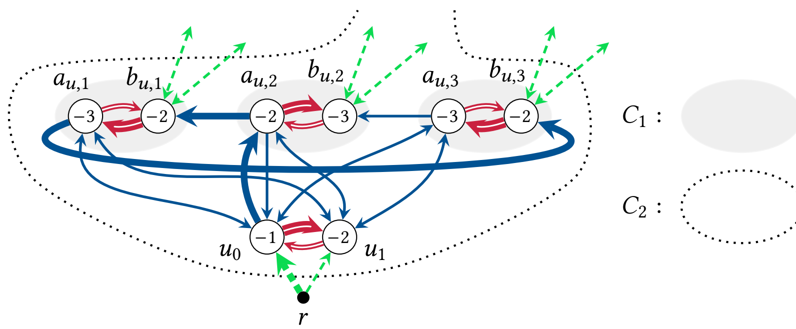


Fig. 3. Illustration of the arborescence A in the proof of Theorem 1.10, shown in bold, together with a feasible dual solution $(\vec{y}, \vec{\alpha})$ certifying $\mu(A) \leq 2n$. The figure assumes $\sigma(u) = 2$. The chain $C_1 \subsetneq C_2 \subsetneq C_3 = E$ is shown using grey and dotted ellipses for edges in C_1 and C_2, respectively, while the values α_v, $v \in V$, are written within the corresponding vertices.

First, assume that there exists some $\mathcal{T} \subseteq \mathcal{S}$ of size n that covers each $u \in U$ exactly once. Let $\sigma(u)$ denote the index in $\{1,2,3\}$ for which $S(u,\sigma(u)) \in \mathcal{T}$. We then let

$$A = \bigcup_{u \in U} \{(r,u_0),(u_0,u_1),(u_0,a_{u,\sigma(u)}),(a_{u,\sigma(u)},b_{u,\sigma(u)})\} \cup (C_u \setminus \{e \in C_u : e \text{ is incident to } b_{u,\sigma(u)}\}).$$

Note that A is an arborescence in G. To prove that the unpopularity margin of A is at most $2n$, we will use the fact that, by definition, $\mu(A) = \max_{A'} \phi(A',A) - \phi(A,A')$ is the optimal value of LP1. Therefore, to show that $\mu(A) \leq 2n$ it suffices to give a dual feasible solution with objective value $2n$. To this end, we define a chain $\mathcal{C} = \{C_1, C_2, C_3\}$ with $C_1 \subsetneq C_2 \subsetneq C_3 = E$ by setting

$$C_1 = \{(a_{u,i},b_{u,i}),(b_{u,i},a_{u,i}) : u \in U, i \in \{1,2,3\}\},$$
$$C_2 = \bigcup_{\{u,v,z\} \in \mathcal{T}} \{e \in E : e \text{ has both endpoints in } V(G_u) \cup V(G_v) \cup V(G_z)\}.$$

Note that $\text{rank}(C_1) = 3|U|$, $\text{rank}(C_2) = (3 \cdot 8 - 1)n = 7|U| + 2n$, and $\text{rank}(C_3) = 8|U|$.

To define a feasible solution $(\vec{y}, \vec{\alpha})$ for LP2, for each $S \subseteq E$ we let $y_S = 1$ if $S \in \mathcal{C}$, and $y_S = 0$ otherwise; we also set

$$\alpha_{a_{u,i}} = \begin{cases} -3 & \text{if } i \neq \sigma(u), \\ -2 & \text{if } i = \sigma(u), \end{cases} \qquad \alpha_{u_0} = -1,$$
$$\alpha_{b_{u,i}} = \begin{cases} -2 & \text{if } i \neq \sigma(u), \\ -3 & \text{if } i = \sigma(u), \end{cases} \qquad \alpha_{u_1} = -2,$$

for each $u \in U$. See Figure 3 for an illustration. The objective value of $(\vec{y}, \vec{\alpha})$ is

$$\sum_{C_i \in \mathcal{C}} \text{rank}(C_i) + \sum_{v \in V} \alpha_v = 3|U| + 7|U| + 2n + 8|U| - 18|U| = 2n.$$

Therefore, to prove that A has unpopularity margin at most $2n$, it suffices to show that $(\vec{y}, \vec{\alpha})$ is a feasible solution for LP2, as stated by Claim 7.1 below.

Claim 7.1. *$(\vec{y}, \vec{\alpha})$ is a feasible solution for LP2.*

Claim proof. We need to verify that

$$|\{C \in \mathcal{C} : e \in C\}| + \alpha_v \geq \text{wt}_A(e) \tag{4}$$

holds for each edge e entering some vertex v. First assume $e \in C_1$, in which case e is contained in three sets of $\mathcal{C}$. If $e \in C_1 \cap A$, then $\mathrm{wt}_A(e) = 0$ and $\alpha_v = -3$ ensures (4). If $e \in C_1 \setminus A$, then $\mathrm{wt}_A(e) = 1$ but $\alpha_v = -2$, so (4) is again satisfied.

Second, assume $e \in C_2 \setminus C_1$, in which case e is contained in two sets from $\mathcal{C}$. If $e \in C_2 \cap A$, then $\alpha_v = -2$, which implies (4). If $e \in C_2 \setminus A$, then we distinguish between two cases: if $v = u_0$ for some $u \in U$, then $\mathrm{wt}_A(e) = 1$ and $\alpha_v = -1$; otherwise $\mathrm{wt}_A(e) = -1$ and $\alpha_v \geq -3$ (note that here we used that all vertices $a_{u,i}$ prefer $(u_0, a_{u,i})$ to $(u_1, a_{u,i})$). Hence, e again satisfies (4).

Third, assume $e \in C_3 \setminus C_2$, in which case e is contained in one set from $\mathcal{C}$. Let G_u be the gadget entered by e. If $e = (r, u_0) \in A$, then $\mathrm{wt}_A(e) = 0$ and $\alpha_v = \alpha_{u_0} = -1$, and thus (4) holds. Otherwise $\mathrm{wt}_A(e) = -1$. Let T be the set in $\mathcal{T}$ containing u. Note that either $v = u_1$ or $v = b_{u,j}$ for some $j \neq \sigma(u)$, because all edges entering $b_{u,\sigma(u)}$ are contained in C_2, since they each originate in some gadget G_z with $z \in T$. Therefore, we have $\alpha_v = -2$ in both cases, which implies (4) for e. ◁

For the other direction, assume that G admits an arborescence A with $\mu(A) \leq 2n$. Let B be an arborescence that yields an optimal solution for LP1, maximizing $\phi(B, A) - \phi(A, B) \leq 2n$. First note that we can assume that A is Pareto-optimal in the sense that there is no arborescence that is weakly preferred by all vertices to A, and strictly preferred by at least one vertex to A. Similarly, we can choose B to be Pareto-optimal as well. Consequently, for any two edges $e, e' \in E_1$ forming a 2-cycle, both A and B uses at least one of e and e'.

For some $X \subseteq V$ and two arborescences A' and A'', let $\phi_X(A', A'')$ denote the number of vertices in X that prefer A' to A''. We say that a gadget G_u is *clean*, if $\phi_{V(G_u)}(B, A) - \phi_{V(G_u)}(A, B) \leq 0$. Let $U^\star = \{u : G_u \text{ is clean}\}$.

Claim 7.2. *If G_u is clean, then a unique edge of A enters G_u, and it comes from r.*

Claim proof. Assume for the sake of contradiction that the claim does not hold for some $u \in U^\star$; this means that A must reach G_u through an edge $e \in E_3$ pointing to some vertex of B_u; let $b_{u,j}$ denote this vertex. Let u_h be the vertex where B enters $\{u_0, u_1\}$; then $(u_h, u_{1-h}) \in B$. Define B' as follows:

$$\begin{aligned} B' = & B \setminus \{\delta(x) : x \in V(G_u)\} \\ & \cup \{(r, u_{1-h}), (u_{1-h}, u_h), (u_0, a_{u,j+1}), (a_{u,j+1}, b_{u,j+1})\} \\ & \cup (C_u \setminus \delta(a_{u,j+1}) \setminus \delta(b_{u,j+1})) \end{aligned}$$

where indices are taken modulo 3 (so $a_{u,4} = a_{u,1}$ and $b_{u,4} = b_{u,1}$).

Observe that B' is an arborescence. If $(u_0, a_{u,j+1}) \notin A$, then $(b_{u,j+1}, a_{u,j+1}) \in A$, and thus vertices u_h, $b_{u,j}$, and $b_{u,j+1}$ all prefer B' to A, while vertices u_{1-h} and $a_{u,j+1}$ prefer A to B'. If $(u_0, a_{u,j+1}) \in A$, then vertices u_h and $b_{u,j}$ prefer B' to A, vertex u_{1-h} prefers A to B', while $a_{u,j+1}$ and $b_{u,j+1}$ are indifferent between them; note $(a_{u,j+1}, b_{u,j+1}) \in A \cap B'$. Furthermore, if $b_{u,j-1}$ prefers A to B', then $(a_{u,j-1}, b_{u,j-1}) \in A$, and therefore $a_{u,j-1}$ prefers B' to A. Summing up all these facts, we obtain $\phi_{V(G_u)}(B', A) - \phi_{V(G_u)}(A, B') \geq 1$ which in turn implies $\phi(B', A) - \phi(A, B') > \phi(B, A) - \phi(A, B)$, a contradiction to our choice of B. ◁

By Claim 7.2, for each $u \in U^\star$ there exists a vertex $\hat{u} \in \{u_0, u_1\}$ for which $(r, \hat{u}) \in A$. We can also assume w.l.o.g. that A and B coincide on G_u, since otherwise we can replace B with the arborescence $B^\star = B \setminus \{\delta(x) : x \in V(G_u), u \in U^\star\} \cup \{A(x) : x \in V(G_u), u \in U^\star\}$, since $B^\star$ is also optimal for LP2. As a consequence, we know that G_u is clean if and only if $\phi_{V(G_u)}(B, A) - \phi_{V(G_u)}(A, B) = 0$.

By $\mu(A) = \phi(B, A) - \phi(A, B) = \sum_{u \in U} \phi_{V(G_u)}(B, A) - \phi_{V(G_u)}(A, B) \leq 2n$, we know that there are at least $|U| - 2n = n$ clean gadgets. Furthermore, we get that for each $u \in U^\star$ there exists some $i \in \{1, 2, 3\}$ for which $A(a_{u,i})$ comes from $\{u_0, u_1\}$; let $\sigma(u)$ denote this index.

Claim 7.3. *If $u \in U^\star$, then the tail of each edge $f \in \delta(b_{u,\sigma(u)}) \cap E_3$ is a descendant of $\hat{u}$ in B.*

Claim proof. Define B_f as follows:

$$B_f = B \setminus \{\delta(x) : x \in V(C_u) \text{ or } x = \hat{u}\} \cup \{f, (a_{u,\sigma(u)}, \hat{u})\} \cup (C_u \setminus \delta(b_{u,\sigma(u)})).$$

Observe that there is an edge from $\hat{u}$ to the other vertex of $\{u_0, u_1\}$ shared by A and B_f.

Suppose that B_f is an arborescence. Note that vertices $\hat{u}$ and $a_{u,\sigma(u)}$ prefer B_f to A, while vertex $b_{u,\sigma(u)}$ prefers A to B_f (because $(a_{u,\sigma(u)}, b_{u,\sigma(u)}) \in A$). Furthermore, if some $b_{u,i}$, $i \neq \sigma(u)$, prefers A to B_f, then $a_{u,i}$ prefers B_f to A. Hence, $\phi_{V(G_u)}(B_f, A) - \phi_{V(G_u)}(A, B_f) \geq 1$, which implies also $\phi(B_f, A) - \phi(A, B_f) > \phi(B, A) - \phi(A, B)$, contradicting our choice of B. Hence, B_f cannot be an arborescence, which can only happen if the tail of f is a descendant of $\hat{u}$ in B. ◁

We claim that $\mathcal{T} = \{S(u, \sigma(u)) : u \in U^\star\}$ is a solution to our instance of Exact 3-Cover. First observe that $\mathcal{T}$ contains at least n sets by $|U^\star| \geq n$. It remains to show that the sets in $\mathcal{T}$ are pairwise disjoint. We say that G_v is *assigned* to $u \in U^\star$, if $v \in S(u, \sigma(u))$. It suffices to show that no gadget G_v can be assigned to more than one vertices in $U^\star$.

Assume for the sake of contradiction that G_v is assigned to both u and w for two distinct vertices $u, w \in U^\star$. Then by Claim 7.3 there are two vertices in B_v, one of them a descendant of $\hat{u}$, the other a descendant of $\hat{w}$. Note that neither $\hat{u}$ nor $\hat{w}$ is a descendant of the other, since both $(r, \hat{u})$ and $(r, \hat{w})$ are edges in A, and hence, in B (recall that A and B coincide on G_u and on G_w). This means that there are two distinct edges entering B_v, one from a descendant of $\hat{u}$, the other from a descendant of $\hat{w}$. Thus for some $j \in \{1, 2, 3\}$, the edges $B(b_{v,j})$ and $B(b_{v,j+1})$ are both in E_3, implying also $(b_{v,j}, a_{v,j}) \in B$ and $(b_{v,j+1}, a_{v,j+1}) \in B$, where indices are taken modulo 3 (so $a_{v,4} = a_{v,1}$ and $b_{v,4} = b_{v,1}$). However, this contradicts the Pareto-optimality of B, since replacing $B(b_{v,j})$ with $(a_{v,j+1}, b_{v,j})$ in B results in an arborescence that $b_{v,j}$ prefers to B, with all other vertices being indifferent between the two.

This shows that any two sets in $\mathcal{T}$ are disjoint, proving the correctness of our reduction.

8 EXTENSIONS AND RELATED RESULTS

Popularity under size constraints. As mentioned in the introduction, the popular largest common independent set problem can be reduced to the popular common base problem. More generally, we can reduce the popular size $[\ell, u]$ common independent set problem to the popular common base problem, where the goal of the former problem is to find a common independent set that is popular within the set of all common independent sets whose size is at least ℓ and at most u (if such a solution exists).

We now describe the reduction. Suppose that we are given a partition matroid M_{part} on $E = \dot\bigcup_{v \in V} E_v$, where each $v \in V$ has a partial order $\succ_v$ over E_v, and an arbitrary matroid $M = (E, \mathcal{I})$. We define a new instance as follows. For each $v \in V$, we create a new element e_v and extend the domain of $\succ_v$ to $E_v \cup \{e_v\}$, where e_v is the unique worst element. The new partition is defined as $E' := \dot\bigcup_{v \in V}(E_v \cup \{e_v\})$. We define a new matroid $M' = (E', \mathcal{I}')$ by

$$\mathcal{I}' = \{\, X \subseteq E' : X \cap E \in \mathcal{I},\ |X \cap E| \leq u,\ |X \cap \{\, e_v : v \in V \,\}| \leq |V| - \ell,\ |X| \leq |V| \,\}.$$

Note that we can assume that the rank of M (i.e., the size of a base in M) is at least ℓ since otherwise the given instance clearly has no solution. Therefore, the rank of M' is $|V|$.

There exists a one-to-one correspondence between common independent sets of sizes in $[\ell, u]$ in the original instance and common bases of the new instance. Suppose I is a common independent set with $\ell \leq |I| \leq u$ in the original instance. Let B be obtained from I by adding e_v for any $v \in V$ with $I \cap E_v = \emptyset$. Then B is a common base in the new instance. Conversely, given a common base of the new instance, we can obtain a common independent set satisfying the size constraint by projecting out the dummy elements. Furthermore, $\phi(I, I') = \phi(B, B')$ holds for any common independent sets I and I' of the original instance and their corresponding bases B and B'. Thus, the reduction is completed.

The reduction used in Section 4 (to reduce the popular common independent set problem to the popular common base problem) is a special case of this reduction where $u = |V|$ and $\ell = 0$.

Popularity under category-wise size constraints. We can also use our popular common base algorithm (Algorithm 1) to solve the problem of finding a common independent set that is popular under a kind of diversity constraints. Such constraints have also been considered in [21] in the context of popular assignments.

Similarly to the above, suppose that a partition matroid on $E = \dot\bigcup_{v \in V} E_v$ with preferences $(\succ_v)_{v \in V}$ and an arbitrary matroid $M = (E, \mathcal{I})$ are given. We regard V as the set of agents. Suppose that the set V is partitioned into q categories $P_1 \dot\cup \cdots \dot\cup P_q$, and each category P_k is associated with integers ℓ_k and u_k where $\ell_k \leq u_k$. We call a common independent set $X \subseteq E$ *admissible* if, for each $k = 1, \ldots, q$, we have $\ell_k \leq |\{ v \in P_k : E_v \cap X \neq \emptyset \}| \leq u_k$. That is, a set X is admissible if, among the agents in each category P_k, at least ℓ_k and at most u_k agents are assigned an element.

The problem of finding a common independent set that is popular within the set of admissible common independent sets can be reduced to the popular common base problem as follows. Similarly to the case of size constraints above, for each $v \in V$, we introduce a dummy element e_v that is worst in v's preference. Moreover, for each category P_k, we create a set D_k of dummy agents with $|D_k| = u_k - \ell_k$. With each dummy agent $d \in D_k$ we associate a set $\{f_d, g_d\}$ of two new elements, and these are tied in the preferences of d, that is, $f_d \not\succ_v g_d$ and $g_d \not\succ_d f_d$. Thus, the new ground set is $E^* = \bigcup_{v \in V}(E_v \cup \{e_v\}) \cup \bigcup_{d \in D_1 \cup \cdots \cup D_q}\{f_d, g_d\}$, and its partition classes are the sets $E_v \cup \{e_v\}$ for $v \in V$ and the sets $\{f_d, g_d\}$ for $d \in D_k$ and $k \in \{1, \ldots, q\}$.

We define a matroid on E^*. First, for $k = 1, \ldots, q$, let $F_k := \{ e_v : v \in P_k \} \cup \{ f_d : d \in D_k \}$ and let $(F_k, \mathcal{I}_k)$ be a uniform matroid defined by $\mathcal{I}_k = \{ X \subseteq F_k : |X| \leq |P_k| - \ell_k \}$.

Next, let $E' := E \cup \{ g_d : d \in D_1 \cup \cdots \cup D_q \}$ and define a matroid $(E', \mathcal{I}')$ as the truncation of the direct sum of M and the free matroid on $\{ g_d : d \in D_1 \cup \cdots \cup D_q \}$, that is,

$$\mathcal{I}' := \left\{ X \subseteq E' : X \cap E \in \mathcal{I}, |X| \leq \textstyle\sum_{k=1}^{q} u_k \right\}.$$

Let $(E^*, \mathcal{I}^*)$ be the direct sum of all these matroids, i.e., $\mathcal{I}^*$ is defined as

$$\mathcal{I}^* = \{ X \subseteq E^* : X \cap E \in \mathcal{I},\ |X \cap E'| \leq \textstyle\sum_{k=1}^{q} u_k,\ |X \cap F_k| \leq |P_k| - \ell_k \text{ for } k = 1, \ldots, q \}.$$

We can assume that the size of a base in $(E, \mathcal{I})$ is at least $\sum_{k=1}^{q} \ell_k$ since otherwise the instance clearly has no admissible set. As we have $|\{ g_d : d \in D_1 \cup \cdots \cup D_q \}| = \sum_{k=1}^{q}(u_k - \ell_k)$, the size of a base in the matroid $(E', \mathcal{I}')$ is exactly $\sum_{k=1}^{q} u_k$. Also, the size of a base in each $(F_k, \mathcal{I}_k)$ is $|P_k| - \ell_k$ (since $|F_k| = |P_k| + u_k - \ell_k$). Thus, the size of a base of the matroid $(E^*, \mathcal{I}^*)$ is $\sum_{k=1}^{q}(|P_k| + u_k - \ell_k)$, which equals the number of agents in the new instance.

We now explain how to transform admissible common independent sets of the original instance to common bases of the new instance, and vise versa. Let I be an admissible common independent set of the original instance. For each $k = 1, \ldots, q$, let $W_k \subseteq P_k$ be the set of agents v in P_k with $I \cap E_v = \emptyset$. Since I is admissible, $|P_k| - u_k \leq |W_k| \leq |P_k| - \ell_k$. Set $B = I$ and augment B by adding elements in the following manner. For each agent v in W_k, add the corresponding element e_v to B. Note that $|P_k| - \ell_k - |W_k|$ is at least 0 and at most $u_k - \ell_k$. Take $|P_k| - \ell_k - |W_k|$ agents d from D_k arbitrarily and add the corresponding f_d elements to B. For the remaining $|D_k| - (|P_k| - \ell_k - |W_k|) = u_k - |P_k| + |W_k|$ agents d in D_k, we add the corresponding g_d elements to B. Thus, all agents are assigned elements. Furthermore, we see that the set B satisfies $B \cap E \in \mathcal{I}$, $|B \cap E'| = |I| + \sum_{k=1}^{q}(u_k - |P_k| + |W_k|) = \sum_{k=1}^{q} u_k$

(note that $\sum_{k=1}^{q}(|P_k|-|W_k|) = |I|$), and $|B\cap F_k| = |P_k|-\ell_k$ for each $k = 1,\dots,q$. Thus, B is a common base in the new instance.

Conversely, let B be a common base of the new instance and I be obtained by deleting all dummy elements in B. Clearly I is a common independent set of the original instance. As B is a base in $\mathcal{I}^*$, we have $|B \cap F_k| = |P_k| - \ell_k$ for each $k = 1,\dots,q$. Since $F_k = \{ e_v : v \in P_k \} \cup \{ f_d : d \in D_k \}$, this implies $|B \cap \{ e_v : v \in P_k \}| \leq |P_k| - \ell_k$. As $|\{ f_d : d \in D_k \}| = u_k - \ell_k$, it also follows that $|B \cap \{ e_v : v \in P_k \}| \geq |P_k| - u_k$. Thus, we have $|P_k| - u_k \leq |B \cap \{ e_v : v \in P_k\}| \leq |P_k| - \ell_k$, which is equivalent to $\ell_k \leq |\{ v \in P_k : B \cap E_v \neq \emptyset \}| \leq u_k$. Thus, I is admissible in the original instance.

We can also observe that $\phi(I,I') = \phi(B,B')$ holds for any admissible common independent sets I and I' of the original instance and their corresponding bases B and B' in the new instance. Therefore, a popular admissible common independent set in the original instance corresponds to a popular common base of the new instance.

Popular fractional solutions. The notion of popularity can be extended to fractional solutions, or equivalently, probability distributions over integral solutions. A fractional/mixed solution x is popular if there is no fractional (in fact, integral) solution more popular than x.

It was shown in [25] using the minimax theorem that popular mixed matchings always exist and such a fractional/mixed matching can be computed in polynomial time. The same proof shows that a popular fractional (largest) common independent set always exists and such a fractional solution can be computed in polynomial time by optimizing over the matroid intersection polytope.

An integral solution I is *strongly popular* if $\phi(I,I') > \phi(I',I)$ for all solutions $I' \neq I$. As observed in [3] in the context of the roommates problem, if a strongly popular solution exists, then it has to be a unique popular fractional solution. Thus there is a polynomial-time algorithm for the strongly popular (largest) common independent set problem.

9 CONCLUSIONS

We considered the popular common base problem, which asks to determine the existence of a popular common base in the intersection of a partition matroid and an arbitrary matroid, where a partial order preference is associated with each partition class of the partition matroid. We provided the first polynomial-time algorithm to solve this problem. This problem includes the popular arborescence problem as a special case, and hence our result affirmatively answers an open problem posed in [26]. Furthermore, we observed that the popular common independent set problem can be reduced to the popular common base problem, and hence can be solved by our algorithm. Utilizing structural observations, we also proved that the min-cost popular common independent set problem is tractable if preferences are weak rankings.

On the intractability side, we proved that the min-cost popular arborescence problem and the k-unpopularity margin arborescence problem are both NP-hard even for strict preferences. Note that the min-cost problem is NP-hard for *popular common bases* (a fact implied by the NP-hardness of the popular assignment problem shown in [21], as well as by Theorem 1.9), while it is tractable for *popular common independent sets* if preferences are weak rankings by Theorem 1.7. By analogy, one may expect the problem of finding a common independent set with unpopularity margin at most k to be polynomial-time solvable. However, this is not the case (unless P = NP), since the k-unpopularity matching problem is NP-hard even for strict rankings [28]. Note that the k-unpopularity margin branching problem is polynomial-time solvable when preferences are weak rankings, as shown in [22], but this does not contradict the above fact: branchings and matchings are both special cases of common independent sets (where one matroid is a partition matroid), but neither of them includes the other. An interesting open question is the following: what is the complexity of finding an arborescence with unpopularity margin at most k, if k is a constant?

ACKNOWLEDGMENTS

We are grateful for inspiring discussions on the popular arborescence problem to Chien-Chung Huang, Satoru Iwata, Tamás Király, Jannik Matuschke, and Ulrike Schmidt-Kraepelin. We thank the anonymous reviewers of the conference version of our paper for their valuable comments. We are grateful to the reviewers of the journal version of our paper for their extremely helpful comments that have improved the presentation. Telikepalli Kavitha is supported by the Department of Atomic Energy, Government of India, under project no. RTI4001. Kazuhisa Makino is partially supported by JSPS KAKENHI Grant Numbers JP20H05967, JP19K22841, and JP20H00609. Ildikó Schlotter is supported by the Hungarian Academy of Sciences under its Momentum Programme (LP2021-2) and its János Bolyai Research Scholarship. Yu Yokoi is supported by JST PRESTO Grant Number JPMJPR212B. This work was supported by JST ERATO Grant Number JPMJER2301, Japan, and the joint project of Kyoto University and Toyota Motor Corporation, titled "Advanced Mathematical Science for Mobility Society."

REFERENCES

[1] David J. Abraham, Robert W. Irving, Telikepalli Kavitha, and Kurt Mehlhorn. 2007. Popular matchings. *SIAM J. Comput.* 37, 4 (2007), 1030–1045. https://doi.org/10.1137/06067328X

[2] Frederick C. Bock. 1971. An algorithm to construct a minimum directed spanning tree in a directed network. In *Developments in Operations Research*, B. Avi-Itzak (Ed.). Gordon and Breach, New York, 29–44.

[3] Felix Brandt and Martin Bullinger. 2022. Finding and recognizing popular coalition structures. *Journal of Artificial Intelligence Research* 74 (2022), 569–626. https://doi.org/10.1613/jair.1.13470

[4] Richard A. Brualdi. 1969. Comments on bases in dependence structures. *Bulletin of the Australian Mathematical Society* 1, 2 (1969), 161–167. https://doi.org/10.1017/S000497270004140X

[5] Yoeng-jin Chu and Tseng-hong Liu. 1965. On the shortest arborescence of a directed graph. *Scientia Sinica* 14 (1965), 1396–1400.

[6] M. Condorcet. 1785. *Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix*. L'Imprimerie Royale.

[7] Ágnes Cseh. 2017. Popular matchings. In *Trends in Computational Social Choice*, Ulle Endriss (Ed.). AI Access, Chapter 6, 105–122. https://real.mtak.hu/80750/1/TrendsCOMSOC_06_u.pdf

[8] Andreas Darmann. 2013. Popular spanning trees. *International Journal of Foundations of Computer Science* 24, 5 (2013), 655 – 677. https://doi.org/10.1142/S0129054113500226

[9] Andreas Darmann. 2016. It is difficult to tell if there is a Condorcet spanning tree. *Mathematical Methods of Operations Research* 84, 1 (2016), 94 – 104. https://doi.org/10.1007/s00186-016-0535-3

[10] Andreas Darmann, Christian Klamler, and Ulrich Pferschy. 2011. Finding socially best spanning trees. *Theory and Decision* 70, 4 (2011), 511 – 527. https://doi.org/10.1007/s11238-010-9228-1

[11] Jack Edmonds. 1967. Optimum branchings. *Journal of Research of the National Institute of Standards* 71B (1967), 233–240. https://doi.org/10.6028/jres.071b.032

[12] Jack Edmonds. 1970. Submodular functions, matroids, and certain polyhedra. In *Combinatorial Structures and Their Applications*, R. Guy, H. Hanani, N. Sauer, and J. Schönheim (Eds.). 69–87.

[13] András Frank. 1981. A weighted matroid intersection algorithm. *Journal of Algorithms* 2 (1981), 328–336. https://doi.org/10.1016/0196-6774(81)90032-8

[14] David Gale and Lloyd S. Shapley. 1962. College admissions and the stability of marriage. *The American Mathematical Monthly* 69, 1 (1962), 9–15. https://doi.org/10.1080/00029890.1962.11989827

[15] Peter Gärdenfors. 1975. Match making: assignments based on bilateral preferences. *Behavioural Science* 20 (1975), 166–173. https://doi.org/10.1002/bs.3830200304

[16] Michel X. Goemans. 2017. Lecture notes on matroid intersection. Massachusetts Institute of Technology, Combinatorial Optimization (Spring 2017), https://math.mit.edu/~goemans/18453S17/matroid-intersect-notes.pdf.

[17] Teofilo F. Gonzalez. 1985. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science* 38 (1985), 293–306. https://doi.org/10.1016/0304-3975(85)90224-5

[18] Steve Hardt and Lia C. R. Lopes. 2015. *Google Votes: A Liquid Democracy Experiment on a Corporate Social Network*. Technical Report. Technical Disclosure Commons. https://www.tdcommons.org/dpubs_series/79

[19] Chien-Chung Huang, Naonori Kakimura, and Naoyuki Kamiyama. 2019. Exact and approximation algorithms for weighted matroid intersection. *Mathematical Programming* 177, 1-2 (2019), 85–112. https://dl.acm.org/doi/10.5555/2884435.2884467

[20] Naoyuki Kamiyama. 2017. Popular matchings with ties and matroid constraints. *SIAM Journal on Discrete Mathematics* 31, 3 (2017), 1801–1819. https://doi.org/10.1137/15M104918X

[21] Telikepalli Kavitha, Tamás Király, Jannik Matuschke, Ildikó Schlotter, and Ulrike Schmidt-Kraepelin. 2022. The popular assignment problem: when cardinality is more important than popularity. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2022)*. SIAM, 103–123. https://doi.org/10.1137/1.9781611977073.6

[22] Telikepalli Kavitha, Tamás Király, Jannik Matuschke, Ildikó Schlotter, and Ulrike Schmidt-Kraepelin. 2022. Popular branchings and their dual certificates. *Mathematical Programming* 192, 1 (2022), 567–595. https://doi.org/10.1007/s10107-021-01659-6

[23] Telikepalli Kavitha, Tamás Király, Jannik Matuschke, Ildikó Schlotter, and Ulrike Schmidt-Kraepelin. 2023. The popular assignment problem: when cardinality is more important than popularity. https://doi.org/10.48550/arXiv.2110.10984 *CoRR*, arXiv:2110.10984 [cs.DS].

[24] Telikepalli Kavitha, Kazuhisa Makino, Ildikó Schlotter, and Yu Yokoi. 2024. Arborescences, colorful forests, and popularity. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2024)*. SIAM, 3724–3746. https://doi.org/10.1137/1.9781611977912.131

[25] Telikepalli Kavitha, Julián Mestre, and Meghana Nasre. 2011. Popular mixed matchings. *Theoretical Computer Science* 412 (2011), 2679–2690. https://doi.org/10.1007/978-3-642-02927-1_48

[26] Tamás Király. 2019. Popular arborescences. 9th Emléktábla Workshop (Matching Theory). https://users.renyi.hu/~emlektab/emlektabla9problems.pdf Page 7.

[27] Mohammad Mahdian. 2006. Random popular matchings. In *Proceedings of the 7th ACM Conference on Electronic Commerce (EC 2006)*. ACM, 238–242. https://doi.org/10.1145/1134707.113473

[28] Richard Matthew McCutchen. 2008. The least-unpopularity-factor and least-unpopularity-margin criteria for matching problems with one-sided preferences. In *Proceedings of the 8th Latin American Conference on Theoretical Informatics (LATIN 2008) (Lecture Notes in Computer Science, Vol. 4957)*. Springer, 593–604. https://doi.org/10.1007/978-3-540-78773-0_51

[29] Samuel Merrill III and Bernard Grofman. 1999. *A Unified Theory of Voting: Directional and Proximity Spatial Models*. Cambridge University Press. https://doi.org/10.1017/CBO9780511605864

[30] Julián Mestre. 2014. Weighted popular matchings. *ACM Transactions on Algorithms* 10, 1 (2014), article 2. https://doi.org/10.1145/2556951

[31] Kei Natsui and Kenjiro Takazawa. 2023. Finding popular branchings in vertex-weighted directed graphs. *Theoretical Computer Science* 953 (2023). https://doi.org/10.1016/j.tcs.2023.113799

[32] Ulrike Schmidt-Kraepelin. 2022. *Models and Algorithms for Scalable Collective Decision Making*. Ph.D. Dissertation. TU Berlin. https://doi.org/10.14279/depositonce-17956

[33] A. Schrijver. 2003. *Combinatorial Optimization - Polyhedra and Efficiency. Vol. 24 of Algorithms and Combinatorics*. Springer-Verlag, Berlin.

[34] Collin T. S. Sng and David F. Manlove. 2010. Popular matchings in the weighted capacitated house allocation problem. *Journal of Discrete Algorithms* 8, 2 (2010), 102–116. https://doi.org/10.1016/j.jda.2008.11.008

[35] Robert Endre Tarjan. 1977. Finding optimum branchings. *Networks* 7, 1 (1977), 25–35. https://doi.org/10.1002/net.3230070103

Appendix A EXAMPLES OF ALGORITHM EXECUTION

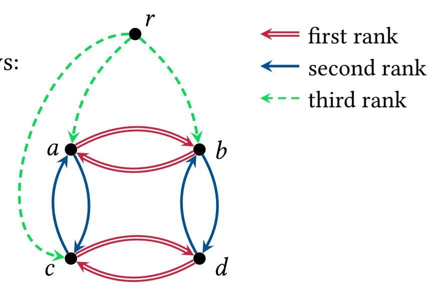
We illustrate how Algorithm 1 works using some examples. We provide three instances of the popular arborescence problem. In all of these instances, a digraph is given as $G = (V \cup \{r\}, E)$ with $V = \{a, b, c, d\}$, and each node $v \in V$ has a strict preference on the set $\delta(v)$ of its incoming edges. For better readability, for a multichain $C = \{C_1, \ldots, C_p\}$ with $C_1 \subseteq \cdots \subseteq C_p$ we will also use the notation $\langle C_1, \ldots, C_p \rangle$.

A.1 Example 1.

This instance is similar to the one illustrated in Section 1; the only difference is that now the edge (r, d) is deleted. In contrast to the case where (r, d) exists, this instance admits a popular arborescence, which is found by Algorithm 1 as follows.

The preference orders for the four vertices are as follows:

$$(b, a) \succ_a (c, a) \succ_a (r, a)$$
$$(a, b) \succ_b (d, b) \succ_b (r, b)$$
$$(d, c) \succ_c (a, c) \succ_c (r, c)$$
$$(c, d) \succ_d (b, d).$$



For convenience, we denote by E_1, E_2, and E_3 the sets of the first, second and third choice edges, respectively. That is, $E_1 = \{(b, a), (a, b), (d, c), (c, d)\}$, $E_2 = \{(c, a), (d, b), (a, c), (b, d)\}$, and $E_3 = \{(r, a), (r, b), (r, c)\}$.

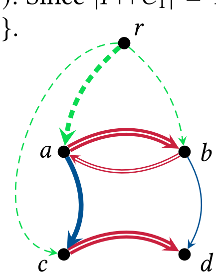
Algorithm Execution. Below we describe the steps in our algorithm.

(1) $p = 1$ and $C_1 = E$. Then $E(C) = E_1$ and $I = \{(a, b), (c, d)\}$ is a lex-maximal branching in $E(C)$. Since $|I \cap C_1| = 2 < 4 = \text{rank}(C_1)$, the set C_1 is updated to $\text{span}(I \cap C_1) = E_1$. Since $C_1 = C_p$ is updated, p is incremented and E is added to C as C_2.

(2) $p = 2$ and $\langle C_1, C_2 \rangle = \langle E_1, E \rangle$. Then $E(C) = E_1 \cup E_2$ and $I = \{(a, b), (c, d), (a, c)\}$ is a lex-maximal branching in $E(C)$. Since $|I \cap C_1| = 2 = \text{rank}(C_1)$ and $|I \cap C_2| = 3 < 4 = \text{rank}(C_2)$, the set C_2 is updated to $\text{span}(I \cap C_2) = E_1 \cup E_2$. Since $C_2 = C_p$ is updated, p is incremented and E is added to C as C_3.

(3) $p = 3$ and $\langle C_1, C_2, C_3 \rangle = \langle E_1, E_1 \cup E_2, E \rangle$. Then $E(C) = \{(c, d)\} \cup E_2 \cup E_3$ and $I = \{(c, d), (c, a), (d, b), (r, c)\}$ is a lex-maximal branching in $E(C)$. Since $|I \cap C_1| = 1 < 2 = \text{rank}(C_1)$, the set C_1 is updated to $\text{span}(I \cap C_1) = \{(c, d), (d, c)\}$.

(4) $p = 3$ and $\langle C_1, C_2, C_3 \rangle = \langle \{(c, d), (d, c)\}, E_1 \cup E_2, E \rangle$. Then $E(C) = \{(r, a), (b, a), (r, b), (a, b), (r, c), (a, c), (c, d), (b, d)\}$ (all edges on the figure to the right) and we have $I = \{(r, a), (a, b), (a, c), (c, d)\}$ (thick edges on the figure to the right) is a lex-maximal branching in $E(C)$. Since $|I \cap C_i| = \text{rank}(C_i)$ holds for $i = 1, 2, 3$, the algorithm returns I.



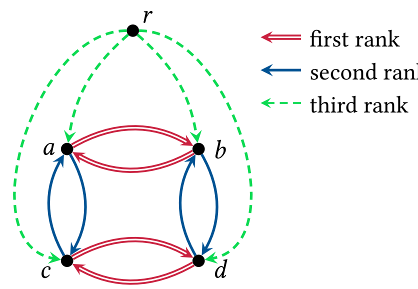
Note that $I' = \{(r, b), (b, a), (a, c), (c, d)\}$ is also a possible output of the algorithm. Indeed, both I and I' are popular arborescences.

A.2 Example 2.

We next demonstrate how the algorithm works for an instance that admits no popular arborescences.

Consider the instance illustrated in the introduction. For the reader's convenience, we include the same figure again. As observed there, this instance has no popular arborescence.

We denote by E_1, E_2, and E_3 the sets of the first, second and third rank edges, respectively. Note that, unlike in Example 1, here E_3 contains (r, d).



Algorithm Execution.

(1) The first step is the same as Step 1 in Example 1. That is, $p = 1$, $C_1 = E$, $E(C) = E_1$, and $I = \{(a, b), (c, d)\}$ is found as a lex-maximal branching in $E(C)$. Then, C_1 is updated to $\text{span}(I \cap C_1) = E_1$, p is incremented, and E is added to C as C_2.

(2) The second step is also the same as Step 2 in Example 1. That is, $p = 2$, $\langle C_1, C_2\rangle = \langle E_1, E\rangle$, $E(C) = E_1 \cup E_2$, and $I = \{(a, b), (c, d), (a, c)\}$ is found as a lex-maximal branching in $E(C)$. Then, C_2 is updated to $\text{span}(I \cap C_2) = E_1 \cup E_2$, p is incremented, and E is added to C as C_3.

(3) $p = 3$ and $\langle C_1, C_2, C_3\rangle = \langle E_1, E_1 \cup E_2, E\rangle$. Then $E(C) = E_2 \cup E_3$ (compared to Example 1, here (r, d) is included while (c, d) is excluded) and $I = \{(a, c), (b, d), (r, a), (r, b)\}$ is a lex-maximal branching in $E(C)$. Since $|I \cap C_1| = 0 < 2 = \text{rank}(C_1)$, the set C_1 is updated to $\text{span}(I \cap C_1) = \emptyset$.

(4) $p = 3$ and $\langle C_1, C_2, C_3\rangle = \langle \emptyset, E_1 \cup E_2, E\rangle$. Then $E(C) = E_1 \cup E_3$ and $I = \{(a, b), (c, d), (r, a), (r, c)\}$ is a lex-maximal branching in $E(C)$. Since $|I \cap C_1| = \text{rank}(C_1)$ and $|I \cap C_2| = 2 < 3 = \text{rank}(C_2)$, the set C_2 is updated to $\text{span}(I \cap C_2) = E_1$.

(5) $p = 3$ and $\langle C_1, C_2, C_3\rangle = \langle \emptyset, E_1, E\rangle$. Then $E(C) = E_1 \cup E_2$ and $I = \{(a, b), (c, d), (a, c)\}$ is a lex-maximal branching in $E(C)$. (Observe that these $E(C)$ and I are the same as Step 2.) Since $|I \cap C_i| = \text{rank}(C_i)$ for $i = 1, 2$ and $|I \cap C_3| = 3 < 4 = \text{rank}(C_3)$, the set C_3 is updated to $\text{span}(I \cap C_3) = E_1 \cup E_2$, p is incremented, and E is added to C as C_4.

(6) $p = 4$ and $\langle C_1, C_2, C_3, C_4\rangle = \langle \emptyset, E_1, E_1 \cup E_2, E\rangle$. Then, as in Step 3, $E(C) = E_2 \cup E_3$ and $I = \{(a, c), (b, d), (r, a), (r, b)\}$ is a lex-maximal branching in $E(C)$. Since $|I \cap C_1| = \text{rank}(C_1)$ and $|I \cap C_2| = 0 < 2 = \text{rank}(C_2)$, the set C_2 is updated to $\text{span}(I \cap C_2) = \emptyset$.

(7) $p = 4$ and $\langle C_1, C_2, C_3, C_4\rangle = \langle \emptyset, \emptyset, E_1 \cup E_2, E\rangle$. By the same argument as in Step 4, the set C_3 is updated to E_1.

(8) $p = 4$ and $\langle C_1, C_2, C_3, C_4\rangle = \langle \emptyset, \emptyset, E_1, E\rangle$. By the same argument as in Step 5, the set C_4 is updated to $E_1 \cup E_2$, p is incremented, and E is added to C as C_5.

(9) $p = 5$ and $\langle C_1, C_2, C_3, C_4, C_5\rangle = \langle \emptyset, \emptyset, E_1, E_1 \cup E_2, E\rangle$. Since $p = 5 > 4 = n = |V|$, the algorithm halts with returning "G has no popular arborescence."

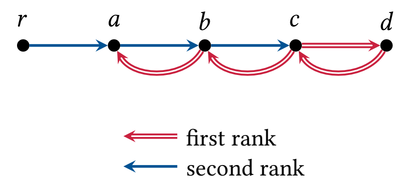
The reader might observe that whenever C_1 becomes empty in the algorithm, then by Lemma 3.3 we can conclude that the instance admits no popular arborescence, since the dual certificate contains only non-empty sets (Lemma 2.2) and hence $D_1 \subseteq C_1 = \emptyset$ is not possible. Therefore, we could in fact stop the algorithm already in Step 3 when C_1 gets updated to $\emptyset$. Nevertheless, the algorithm will reach a correct answer even without using this observation, as illustrated by the above example.

A.3 Example 3.

We next provide an example that shows the importance of considering multichains. During the algorithm's execution on this instance, C does become a multichain that is not a chain.

The preferences of the four vertices are as follows:

$$(b,a) \succ_a (r,a)$$
$$(c,b) \succ_b (a,b)$$
$$(d,c) \succ_c (b,c)$$
$$(c,d)$$

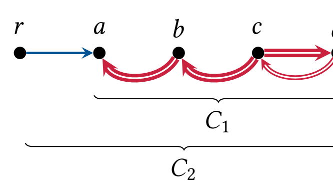


where (c,d) is the unique incoming edge of d. For convenience, we denote by E_{abcd}, E_{bcd}, and E_{cd} the edge sets of the induced subgraphs for the vertex sets $\{a,b,c,d\}$, $\{b,c,d\}$, and $\{c,d\}$, respectively. That is, $E_{abcd} = E \setminus \{(r,a)\}$, $E_{bcd} = \{(b,c),(c,b),(c,d),(d,c)\}$, and $E_{cd} = \{(c,d),(d,c)\}$. Note that $\{(r,a),(a,b),(b,c),(c,d)\}$ is the unique arborescence in this instance, and hence it is a popular arborescence.

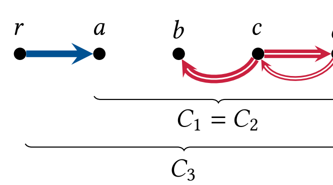
Algorithm Execution.

(1) $p = 1$ and $C_1 = E$. Then $E(C) = \{(b,a),(c,b),(d,c),(c,d)\}$ and $I = \{(b,a),(c,b),(c,d)\}$ is a lex-maximal branching in $E(C)$. Since $|I \cap C_1| = 3 < 4 = \text{rank}(C_1)$, the set C_1 is updated to $\text{span}(I \cap C_1) = E_{abcd}$. Since $C_1 = C_p$ is updated, p is incremented and E is added to C as C_2.

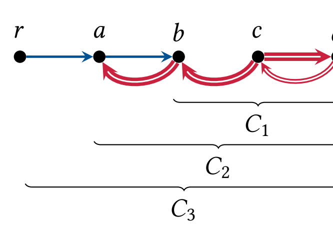
(2) $p = 2$ and $\langle C_1, C_2\rangle = \langle E_{abcd}, E\rangle$ (shown by braces on the right). Then $E(C) = \{(r,a),(b,a),(c,b),(d,c),(c,d)\}$ (all edges on the right) and $I = \{(b,a),(c,b),(c,d)\}$ (thick edges on the right) is a lex-maximal branching in $E(C)$. Since $|I \cap C_1| = \text{rank}(C_1)$ and $|I \cap C_2| = 3 < 4 = \text{rank}(C_2)$, C_2 is updated to $\text{span}(I \cap C_2) = E_{abcd}$. Since $C_2 = C_p$ is updated, p is incremented and E is added to C as C_3.



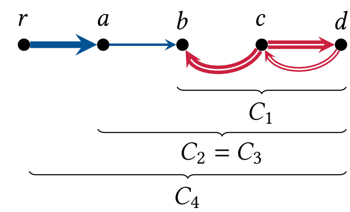
(3) $p = 3$ and $\langle C_1, C_2, C_3\rangle = \langle E_{abcd}, E_{abcd}, E\rangle$ (so $C_1 = C_2$). Then $E(C) = \{(r,a),(c,b),(d,c),(c,d)\}$. Note that (b,a) is not in $E(C)$ as $\text{lev}_C((b,a)) = 1$ while $\text{lev}_C((r,a)) = 3$. $I = \{(r,a),(c,b),(c,d)\}$ is a lex-maximal branching in $E(C)$. Since $|I \cap C_1| = 2 < 3 = \text{rank}(C_1)$, the set C_1 is updated to $\text{span}(I \cap C_1) = E_{bcd}$.



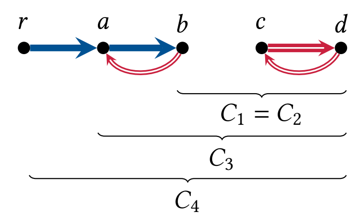
(4) $p = 3$ and $\langle C_1, C_2, C_3\rangle = \langle E_{bcd}, E_{abcd}, E\rangle$. Then, $E(C) = E \setminus \{(b,c)\}$ and $I = \{(b,a),(c,b),(c,d)\}$ is a lex-maximal branching in $E(C)$. Since $|I \cap C_i| = \text{rank}(C_i)$ for $i = 1,2$ and $|I \cap C_3| = 3 < 4 = \text{rank}(C_3)$, the set C_3 is updated to $\text{span}(I \cap C_3) = E_{abcd}$. Since $C_3 = C_p$ is updated, p is incremented and E is added to C as C_4.



(5) $p = 4$ and $\langle C_1, C_2, C_3, C_4 \rangle = \langle E_{bcd}, E_{abcd}, E_{abcd}, E \rangle$. Then $E(C) = E \setminus \{(b,a),(b,c)\}$ and $I = \{(r,a),(c,b),(c,d)\}$ is a lex-maximal branching in $E(C)$. Since $|I \cap C_1| = \text{rank}(C_1)$ and $|I \cap C_2| = 2 < 3 = \text{rank}(C_2)$, the set C_2 is updated to $\text{span}(I \cap C_2) = E_{bcd}$.



(6) $p = 4$ and $\langle C_1, C_2, C_3, C_4 \rangle = \langle E_{bcd}, E_{bcd}, E_{abcd}, E \rangle$. Then $E(C) = E \setminus \{(b,c),(c,b)\}$ and $I = \{(r,a),(a,b),(c,d)\}$ is a lex-maximal branching in $E(C)$. Since $|I \cap C_1| = 1 < 2 = \text{rank}(C_1)$, the set C_1 is updated to $\text{span}(I \cap C_1) = E_{cd}$.



(7) $p = 4$ and $\langle C_1, C_2, C_3, C_4 \rangle = \langle E_{cd}, E_{bcd}, E_{abcd}, E \rangle$. Then $E(C) = E$ and $I = \{(r,a),(a,b),(b,c),(c,d)\}$ is a lex-maximal branching in $E(C)$. Since $|I \cap C_i| = \text{rank}(C_i)$ holds for $i = 1,2,3,4$, the algorithm returns I.