Mathematical Programming, Series A (2017) 161:347–364
DOI 10.1007/s10107-016-1011-9

Security Games on Matroids

Dávid Szeszlér

Received: 20 July 2015 / Accepted: 1 April 2016

Abstract Two players, the Defender and the Attacker play the following game. A matroid $M = (S, \mathscr{I})$, a weight function $d : S \to \mathbb{R}^+$ and a cost function $c : S \to \mathbb{R}$ are given. The Defender chooses a base B of the matroid M and the Attacker chooses an element $s \in S$ of the ground set. In all cases, the Attacker pays the Defender the cost of attack $c(s)$. Besides that, if $s \in B$ then the Defender pays the Attacker the amount $d(s)$; if, on the other hand, $s \notin B$ then there is no further payoff. Special cases of this two-player, zero-sum game were considered and solved in various security-related applications. In this paper we show that it is also possible to compute Nash-equilibrium mixed strategies for both players in strongly polynomial time in the above general matroid setting. We also consider a further generalization where common bases of two matroids are chosen by the Defender and apply this to define and efficiently compute a new reliability metric on digraphs.

Keywords Nash Equilibrium · Matroid · Security

Mathematics Subject Classification (2000) MSC 90C27 · MSC 91A80

Research supported by grant No. OTKA 185101 of the Hungarian Scientific Research Fund.

D. Szeszlér
Department of Computer Science and Information Theory
Budapest University of Technology and Economics
Magyar Tudósok Körútja 2., Budapest, 1117, Hungary
E-mail: szeszler@cs.bme.hu

1 Introduction

There is an abundance of recent books and papers on game-theoretical tools for measuring and increasing security. Since all aspects of security are obviously of utmost importance nowadays and game theory as a tool to address related problems presents itself very naturally, the literature on this topic is extremely diverse. Much of the arsenal of game theory has been employed on various applications which very often have little in common besides somehow being related to security. Interested readers are referred to the following books and surveys: [1,7,9,10,14].

In this paper, however, only the theory of two player, zero-sum games, the simplest and probably most widely known subfield of game theory will be relied on to address various problems raised by applications concerning the measuring of security. The basic idea is very natural: define a game between two virtual players, the Attacker and the Defender, such that the rules of the game and the payoffs to be paid capture the circumstances under which security is to be measured. Then analyzing the game might give rise to an appropriate security metric: the better the Attacker can do in the game, the lower the level of security is. If the game is zero-sum then the maximum guaranteed expected gain the Attacker can achieve (by a mixed strategy) is equal to the minimum guaranteed expected loss the Defender has to suffer by Neumann's classic Minimax Theorem [12]; hence the reciprocal of this common optimum is a valid measure of security.

The following simple example might illuminate the above idea. Assume that a connected graph G and two vertices $s,t \in V(G)$ are given. The Defender chooses a path P between s and t and the Attacker (simultaneously) chooses an edge e of G. If e is not on P then there is no payoff; if, on the other hand, e is on P then the Defender pays 1 to the Attacker. Then it is easy to see that the Nash-equilibrium payoff of this two-player, zero-sum game is the reciprocal of the edge-connectivity between s and t (that is, the maximum number of pairwise edge-disjoint paths between s and t or, equivalently by Menger's classic theorem [13, Section 9.1], the size of the minimum cut separating s and t). In other words, the notion of edge-connectivity between two vertices (viewed as a security metric) is well captured by this simple game.

In [5] the following *Spanning Tree Game* is considered. Given a connected graph G and a cost function on its edges $c : E(G) \to \mathbb{R}$, the Defender chooses a spanning tree T of G (that can be viewed as some communication infrastructure), while the Attacker chooses (or "attacks") an edge $e \in E(G)$. Then the payoff from the Defender to the Attacker is $1 - c(e)$ if e is in T and $-c(e)$ otherwise. As the main result of [5] a formula is presented for the Nash-equilibrium payoff of this game (which also implies a description of Nash-equilibrium mixed strategies for the Attacker). Besides allowing for a more general, matroidal setting, we will generalize the results of [5] in the following ways. Firstly, we allow that the Attacker's gain depends on the chosen edge e: it will be $d(e) - c(e)$ if e is in T (and $-c(e)$ otherwise), where $d : E(G) \to \mathbb{R}^+$ is a given positive weight function. Secondly, we show that the Nash-equilibrium payoff of the game, as well as optimum mixed strategies for both players are computable in strongly polynomial time. We will also discuss a version of this game on digraphs.

It is also known that in the $c(e) \equiv 0$ and $d(e) \equiv 1$ case of the Spanning Tree Game the Nash-equilibrium payoff is equal to the reciprocal of another interesting graph reliability metric: the *strength* of a connected graph G, as defined by Gusfield [6], is $\sigma(G) = \min\left\{\frac{|U|}{\mathrm{comp}(G-U)-1} : U \subseteq E(G), \mathrm{comp}(G-U) > 1\right\}$, where $\mathrm{comp}(G-U)$ is the number of components of the graph obtained from G by deleting U. The equality between $\sigma(G)$ and the reciprocal of the Nash-equilibrium payoff of the Spanning Tree Game is stated and proved in [4], however, an equivalent result (in a non-game-theoretical setting) was derived in [2] from the classic edge-disjoint spanning trees theorem of Nash-Williams and Tutte [13, Corollary 51.1a]. The notion of graph strength was extended to a weighted version by Cunningham [2]:

$$\sigma_p(G) = \min\left\{\frac{p(U)}{\mathrm{comp}(G-U)-1} : U \subseteq E(G), \mathrm{comp}(G-U) > 1\right\}, \tag{1}$$

where $p : E(G) \to \mathbb{R}^+$ is a positive weight function. (Here $p(U) = \sum\{p(e) : e \in U\}$; we use this notation throughout.) In [2] a strongly polynomial algorithm was also given for computing $\sigma_p(G)$. It will follow from the results of this paper (see Theorem 5) that $\sigma_p(G)$ is the reciprocal of the Nash-equilibrium payoff of the Spanning Tree Game if $c(e) \equiv 0$ and $d(e) = \frac{1}{p(e)}$ for all e.

In [8] a similar looking, but essentially different game was considered, motivated by an application in measuring the security of content-adaptive steganography. (Steganography is the science of hiding a message in a cover file and content-adaptive steganography is a subfield of this area that is sensitive to the varying predictability of different parts of a cover file; see [8] for the details.) There the Defender chooses any k-element subset H of an n-element ground set S and the Attacker chooses an element $s \in S$. Then the payoff from the Defender to the Attacker is $d(s) - c(s)$ if $s \in H$ and $-c(s)$ otherwise. In [8] a strongly polynomial algorithm was given for solving this game – meaning that the Nash-equilibrium payoff and optimum mixed strategies for both players are computed. (In fact, the algorithm of [8] is not only strongly polynomial, it is efficient even for problem instances corresponding to the sizes of steganography-related applications, where the magnitude of n can easily be in the tens of thousands; see the details in [8].)

Since spanning trees and k-element subsets are the bases of the cycle matroid of a connected graph and the uniform matroid, respectively, the following definition gives a common generalization of the above games. It also contains, for example, the generalizations of the above mentioned Spanning Tree Game where spanning edge sets of a given size or the unions of edge sets of a given number of pairwise edge-disjoint spanning trees are chosen by the Defender (as these are the bases of the matroid sums of the cycle matroid and a uniform matroid or copies of the cycle matroid, respectively).

Definition 1 Assume that a matroid $M = (S, \mathscr{I})$, a positive valued weight function $d : S \to \mathbb{R}^+$ and a real valued cost function $c : S \to \mathbb{R}$ are given. The *Matroid Base Game* is played by two players, the Attacker and the Defender. The Attacker chooses an element $s \in S$ and (simultaneously) the Defender chooses a base B of M. Then the payoff paid by the Defender to the Attacker is $d(s) - c(s)$ or $-c(s)$, if $s \in B$ or $s \notin B$, respectively. (A negative payoff obviously means in practice that it is the Attacker that pays the Defender the absolute value of the payoff.)

We remark that it is a sensible assumption made in the definition of the above mentioned security-related applications that the cost of attack $c(s)$ paid by the Attacker should not be received by the Defender (as the Defender is indifferent to the costs and efforts associated with an attack, she is only affected by the damage caused). In other words, the payoffs given in the above definition should only correspond to the Attacker while Defender's loss should be $d(s)$ or 0 if $s \in B$ or $s \notin B$, respectively. This would also imply that the game is not zero-sum any more. However, it is easily shown that the thus-obtained non-zero-sum game is essentially equivalent to the zero-sum game defined in Definition 1. This equivalency is due to the fact that the sum of the payoffs only depends on the choice of the Attacker and it more precisely means that Nash-equilibria of the two versions of the game are identical and the Attacker's Nash-equilibrium payoff is unique in the non-zero-sum version of the game and it is equal to the (unique) Nash-equilibrium payoff corresponding to the zero-sum version. (An analogous statement would not be true for the Defender.) The proof of this equivalency is a simple exercise that seems to be folklore (see [8, Lemma 1] for a proof). We will disregard this point in the remainder of the paper and focus on the zero-sum game defined above.

We also disregard the detail that the Attacker is granted the right to refrain from the attack (that is, reject participating in the game) in the above security-related applications by observing that obviously, this is the rational decision for the Attacker if and only if his Nash-equilibrium payoff (to be determined in general) is non-positive.

This paper is structured as follows. In Sect. 2 we list some necessary preliminary results. In Sect. 3 we first derive some basic structural properties of an optimum solution of the Matroid Base Game and then use these to present a strongly polynomial algorithm to solve the game. In Sect. 4 we consider the generalization of the problem where the Defender chooses a common base of two matroids. Although the algorithm of Sect. 3 does not generalize to this case, we will give some partial results. These will enable us to discuss a directed analogue of the above mentioned notion of graph strength (1) and the corresponding Spanning Tree Game.

The main result of this paper is the strongly polynomial algorithm of Sect. 3 that solves the Matroid Base Game in the sense that it computes the Nash-equilibrium payoff and an optimum mixed strategy for both players (see Theorem 7). It should be noted that the existence of such an algorithm is already known in an important special case: it will follow from the results of Sect. 3.1 that if $c \equiv 0$ is assumed then the problem is equivalent to the *capacitated fractional base packing* problem which is discussed and solved in [13, Section 42.4]. Therefore the main contribution of this paper is the generalization of that result to the case of an arbitrary cost function. This generalization is non-trivial: in Sect. 4.1 we will present a version of the algorithm of [13, Section 42.4] adapted to the case where common bases of two matroids are considered and we will see that further constraints on the input data seem to be necessary in order to prove a strongly polynomial running time (see Proposition 16). These constraints are trivially fulfilled in the $c \equiv 0$ case, but not for a general c and d – and the same problem would present itself if the algorithm of [13, Section 42.4] were applied on the original game of Definition 1.

We follow the notation and terminology of [13]. All necessary background on matroid theory and polyhedra is also to be found in [13]. We also use [13] as a source of citations for results considered to be classic and widely known. (In some other cases, citations from [13] point to results that seem to be new in [13].) From game theory we only rely on the basics of two-player, zero-sum games covered by many textbooks on linear programming, see [11] for example.

2 Preliminary results

We will greatly rely on the descriptions of certain basic polyhedra associated with matroids. Given a matroid $M=(S,\mathscr{I})$, the *independent set polytope* $P_{\text{independent set}}(M)$ and the *base polytope* $P_{\text{base}}(M)$ are defined as the convex hulls of incidence vectors of independent sets and bases, respectively. The rank function of the matroid $M=(S,\mathscr{I})$ is denoted by r throughout the paper. The following theorem is due to Edmonds [13, Corollaries 40.2c and 40.2d].

Theorem 1

$$\begin{aligned}P_{\textit{independent set}}(M)=\big\{x\in\mathbb{R}^S : & \, x(U)\le \mathrm{r}(U) \textit{ for all } U\subseteq S,\\ & x(s)\ge 0 \textit{ for all } s\in S\big\}\end{aligned}$$

and

$$\begin{aligned}P_{\textit{base}}(M)=\big\{x\in\mathbb{R}^S : & \, x(U)\le \mathrm{r}(U) \textit{ for all } U\subseteq S,\\ & x(S)=\mathrm{r}(S),\\ & x(s)\ge 0 \textit{ for all } s\in S\big\}.\end{aligned}$$

The following minimax theorem, also due to Edmonds, is an easy corollary of the above description of $P_{\text{independent set}}(M)$ (see [13, Theorem 40.3] for a one-paragraph proof).

Theorem 2 *Let $M=(S,\mathscr{I})$ be a matroid and $z\in\mathbb{R}^S$, $z\ge 0$ an arbitrary non-negative vector. Then*

$$\max\{x(S) : x\in P_{\textit{independent set}}(M), x\le z\}=\min\{\mathrm{r}(U)+z(S-U) : U\subseteq S\}.$$

The *up-hull* of a polyhedron $P\subseteq\mathbb{R}^S$ is defined as $P^{\uparrow}=\{z\in\mathbb{R}^S : \exists x\in P, x\le z\}$. In other words, $P^{\uparrow}$ is the Minkowski-sum of P and the non-negative orthant of $\mathbb{R}^S$. The above theorem immediately implies the following description of the up-hull of the base polytope since $z\in P^{\uparrow}_{\text{base}}(M)$ if and only if the maximum in Theorem 2 is $\mathrm{r}(S)$.

Corollary 3 $P^{\uparrow}_{\textit{base}}(M)=\big\{x\in\mathbb{R}^S : x(S-U)\ge\mathrm{r}(S)-\mathrm{r}(U) \textit{ for all } U\subseteq S\big\}$.

The above polyhedral results raise algorithmic questions: can one efficiently decide if a given vector belongs to any of the above polyhedra? All these are answered in the affirmative by Cunningham's algorithm [3] that approaches Theorem 2 algorithmically. Observe that for an arbitrary $x\in P_{\text{independent set}}(M)$, $x\le z$ and $U\subseteq S$ the following holds: $x(S)=x(U)+x(S-U)\le\mathrm{r}(U)+z(S-U)$. This shows the max $\le$ min relation in Theorem 2 and also that $x\in P_{\text{independent set}}(M)$ and $U\subseteq S$ are optimal for Theorem 2 if and only if $x(U)=\mathrm{r}(U)$ and $x(s)=z(s)$ for all $s\in S-U$. In other words, Theorem 2 claims the existence of such a pair of x and U, but Cunningham's algorithm also finds them efficiently.

Theorem 4 **([3])** *Assume that a matroid $M=(S,\mathscr{I})$ is given by an independence testing oracle and a vector $z\in\mathbb{Q}^S$, $z\geq 0$ is also given. Then there exists a strongly polynomial algorithm that computes a vector $x\in P_{independent\ set}(M)$, $x\leq z$ and a subset $U\subseteq S$ such that $x(U)=\mathrm{r}(U)$ and $x(s)=z(s)$ for all $s\in S-U$ hold. Furthermore, the algorithm also computes a decomposition of x as a convex combination of incidence vectors of independent sets of M.*

It follows immediately that by Cunningham's algorithm one can test the membership of any given vector in $P_{\text{independent set}}(M)$, $P_{\text{base}}(M)$ or $P^{\uparrow}_{\text{base}}(M)$. A nice description of the algorithm is given in [13, Theorem 40.4].

3 The Matroid Base Game

In this section we solve the Matroid Base Game. We start with some non-algorithmic results and then, using these, we provide a strongly polynomial algorithm.

3.1 Non-algorithmic results

Theorem 5 *Assume that the matroid $M=(S,\mathscr{I})$ and the vectors $d,c\in\mathbb{R}^S$, $d>0$ are given. Then the Nash-equilibrium payoff of the Matroid Base Game is equal to*

$$\min\left\{\mu:\mu\cdot p+q\in P^{\uparrow}_{base}(M)\right\}=\max_{U\subseteq S, U\neq S}\frac{\mathrm{r}(S)-\mathrm{r}(U)-q(S-U)}{p(S-U)},$$

where $p(s)=\frac{1}{d(s)}$ and $q(s)=\frac{c(s)}{d(s)}$ for all $s\in S$.

Proof Denote the set of bases of M by $\mathscr{B}$ and assume that a mixed strategy of the Defender $\{\delta(B):B\in\mathscr{B}\}$ is given. In other words, δ is a probability distribution on $\mathscr{B}$. Then assuming that the Attacker chooses a given fixed element $s\in S$ in the game, the Defender's expected loss is

$$\sum_{s\in B\in\mathscr{B}}\delta(B)\cdot\big(d(s)-c(s)\big)-\sum_{s\notin B\in\mathscr{B}}\delta(B)\cdot c(s)=d(s)\cdot\left(\sum_{s\in B\in\mathscr{B}}\delta(B)\right)-c(s). \quad (2)$$

Let $x(s)=\sum\{\delta(B):s\in B\in\mathscr{B}\}$ for all $s\in S$. Then the vector $x\in\mathbb{R}^S$ is nothing but an element of $P_{\text{base}}(M)$ by definition (since the values $\delta(B)$ form the set of coefficients of a convex combination). Since, by definition, the Defender's objective is to minimize the maximum expected loss she has to suffer, her task amounts to the following by (2):

$$\min\left\{\mu:\exists x\in P_{\text{base}}(M), d(s)\cdot x(s)-c(s)\leq\mu \text{ for all } s\in S\right\}.$$

Rearranging this gives that the Defender's objective is equivalent to the following:

$$\min\left\{\mu:\exists x\in P_{\text{base}}(M), x\leq\mu\cdot p+q\right\}.$$

Using the definition of $P^{\uparrow}_{\text{base}}(M)$ this is further equivalent to the following:

$$\min\{\mu : \mu \cdot p + q \in P^{\uparrow}_{\text{base}}(M)\}. \tag{3}$$

This, together with Neumann's Minimax Theorem [12] already proves that the Nash-equilibrium payoff of the game is equal to the minimum in the theorem. However, by Corollary 3, $\mu \cdot p + q \in P^{\uparrow}_{\text{base}}(M)$ is true if and only if

$$\mu \cdot p(S-U) + q(S-U) \geq \mathrm{r}(S) - \mathrm{r}(U)$$

holds for all $U \subseteq S$. Then simple rearranging (and observing that this inequality is trivial for $U = S$) immediately gives that $\mu \cdot p + q \in P^{\uparrow}_{\text{base}}(M)$ is true if and only if μ is greater than or equal to the maximum on the right hand side of the equation in the theorem. Hence the minimum of all such μ's is exactly this maximum. □

Since the minimum in Theorem 5 corresponds to the minimum expected loss the Defender has to suffer and that, by Neumann's Minimax Theorem, is equal to the Attacker's maximum guaranteed expected gain, it is not much of a surprise that the maximum in Theorem 5 is related to the latter.

Proposition 6 *Assume that the subset $U \subseteq S$ maximizes the right hand side of the equation in Theorem 5 and p is defined as in Theorem 5. Then*

$$\alpha(s) = \begin{cases} \dfrac{p(s)}{p(S-U)} & \text{if } s \in S-U \\ 0 & \text{if } s \in U \end{cases}$$

defines an optimum mixed strategy for the Attacker in the Matroid Base Game.

Proof Denote by μ^* the Nash-equilibrium payoff of the game which is equal to the common optimum in Theorem 5. We need to show that no matter which base the Defender chooses, α guarantees the Attacker an expected gain of at least μ^*. So assume the Defender chooses the base B; then the Attacker's expected gain is

$$\sum_{s\in B-U} \frac{p(s)}{p(S-U)} \cdot \big(d(s) - c(s)\big) - \sum_{s\in S-U-B} \frac{p(s)}{p(S-U)} \cdot c(s) =$$

$$\sum_{s\in B-U} \frac{p(s)}{p(S-U)} \cdot d(s) - \sum_{s\in S-U} \frac{p(s)}{p(S-U)} \cdot c(s) =$$

$$\frac{|B-U|}{p(S-U)} - \frac{q(S-U)}{p(S-U)} \geq \frac{\mathrm{r}(S) - \mathrm{r}(U) - q(S-U)}{p(S-U)} = \mu^*,$$

where we used $p(s) \cdot d(s) = 1$, $p(s) \cdot c(s) = q(s)$ and $|B-U| \geq \mathrm{r}(S) - \mathrm{r}(U)$. (The latter is true since $|B| = \mathrm{r}(S)$ and $|B \cap U| \leq \mathrm{r}(U)$.) □

The above proposition implies, for example, the slightly surprising fact that if $d(s) = 1$ for all $s \in S$ (and c is arbitrary) then the uniform distribution on a suitably chosen subset $U \subseteq S$ is an optimal mixed strategy for the Attacker.

Theorem 5 and Proposition 6 together already contain and generalize most results of [5] on the Spanning Tree Game mentioned in Sect. 1 (although in [5] the results are stated in a somewhat more extensive form and the proofs are much more lengthy and complicated).

3.2 A Strongly polynomial algorithm

As it is covered by many introductory textbooks on linear programming (see [11, Section 8.1] for example), every two-player zero-sum game given by its payoff matrix is solvable in polynomial time via linear programming. Obviously, this is not a viable option in case of the Matroid Base Game since the number of the Defender's possible choices (that is, the number of bases of M) is typically exponential in $|S|$ which makes the size of the payoff matrix also exponential. The following theorem is the main contribution of the paper.

Theorem 7 *Assume that a matroid $M = (S, \mathscr{I})$ is given by an independence testing oracle and the vectors $d, c \in \mathbb{Q}^S$, $d > 0$ are also given. Then there exists a strongly polynomial algorithm that computes the Nash-equilibrium payoff of the Matroid Base Game and an optimum mixed strategy for both players.*

As already mentioned in Sect. 1, the above theorem is essentially known in the special case of $c = 0$: then, by the proof of Theorem 5, solving the Matroid Base Game is equivalent to the *capacitated fractional base packing* problem discussed in [13, Section 42.4], where a strongly polynomial algorithm is given in [13, Theorem 42.7]. However, that algorithm does not seem to generalize to the $c \neq 0$ case. Hence the above theorem can also be regarded as a generalization of [13, Theorem 42.7].

Proof of Theorem 7. Denote p and q as in Theorem 5. By Theorem 5, we need to compute a value μ and a subset $U \subseteq S$, $U \neq S$ that are optimal for the minimax relation in Theorem 5. Indeed, then μ is the Nash-equilibrium payoff of the game, an optimum mixed strategy for the Attacker is given by U according to Proposition 6 and running Cunningham's algorithm (Theorem 4) on $z = \mu \cdot p + q$ yields a decomposition of an $x \in P_{\text{base}}(M)$, $x \leq z$ as a convex combination of incidence vectors of bases, the coefficients of which describe an optimum mixed strategy for the Defender according to the proof of Theorem 5.

(Note that $x \in P_{\text{base}}(M)$ holds for the x computed by Cunningham's algorithm. Indeed, according to Theorem 2, x maximizes $x(S)$ over all vectors for which $x \in P_{\text{independent set}}(M)$ and $x \leq z$ hold. But since $z \in P^{\uparrow}_{\text{base}}(M)$ implies the existence of such an x with $x(S) = \mathrm{r}(S)$, the one computed by the algorithm must be in $P_{\text{base}}(M)$. This also implies that in the convex combination computed by the algorithm all incidence vectors correspond to bases.)

Let $\mu \in \mathbb{R}$, $U \subseteq S$ and $x \in P_{\text{base}}(M)$ be arbitrary such that $x \leq \mu \cdot p + q$. Then

$$\mathrm{r}(S) = x(S) = x(U) + x(S-U) \leq \mathrm{r}(U) + \mu \cdot p(S-U) + q(S-U) \tag{4}$$

follows from the description of $P_{\text{base}}(M)$ (Theorem 1). Rearranging this gives $\mu \geq \frac{\mathrm{r}(S)-\mathrm{r}(U)-q(S-U)}{p(S-U)}$ that is, the max $\leq$ min relation of Theorem 5. Furthermore, the optimality conditions for Theorem 5 can also be extracted from (4): $\mu = \frac{\mathrm{r}(S)-\mathrm{r}(U)-q(S-U)}{p(S-U)}$ holds if and only if $x(U) = \mathrm{r}(U)$ and $x(s) = \mu \cdot p(s) + q(s)$ for all $s \in S - U$.

The algorithm will maintain a value μ, a subset $U \subseteq S$, $U \neq S$ and a vector $z \in \mathbb{R}^S$ such that the following conditions are met throughout:

1. $0 \leq z \leq \mu \cdot p + q$
2. $z(s) = \mu \cdot p(s) + q(s)$ for all $s \in S - U$
3. $z(U) = \mathrm{r}(U)$
4. $z(S) = \mathrm{r}(S)$
5. $z(W) \leq \mathrm{r}(W)$ for all subsets $W \subseteq U$

In other words, almost all optimality conditions are maintained with the single exception being that $z \in P_{\text{base}}(M)$ is relaxed by not prescribing $z(W) \leq \mathrm{r}(W)$ on subsets $W \not\subseteq U$. The algorithm terminates when $z \in P_{\text{base}}(M)$ is achieved.

Initializing the algorithm is not at all trivial, we will come back to this at the end of the proof. Instead, we assume for now that μ, z and U fulfill conditions 1–5 and describe the steps the algorithm keeps iterating:

Algorithm 1.

Step 1. Run Cunningham's algorithm (Theorem 4) on z. Assume it gives $x \in P_{\text{independent set}}(M)$ and $Y \subseteq S$ such that

$$x \leq z, x(Y) = \mathrm{r}(Y) \text{ and } x(s) = z(s) \text{ for all } s \in S - Y. \tag{5}$$

If $x = z$ then STOP and output μ, U (and x).

Step 2. If $x \neq z$ then let

$$\begin{aligned} U' &:= U \cup Y, \\ \mu' &:= \frac{\mathrm{r}(S) - \mathrm{r}(U') - q(S - U')}{p(S - U')} \text{ and} \\ z'(s) &:= \begin{cases} x(s) & \text{if } s \in U' \\ \mu' \cdot p(s) + q(s) & \text{if } s \in S - U'. \end{cases} \end{aligned}$$

Continue at *Step 1* with U', μ' and z' instead of U, μ and z.

We will show that these steps maintain conditions 1–5. This is immediately true for conditions 2 and 5: the former by the definition of z' and the latter because $z'(s) = x(s)$ for all $s \in U'$ and $x \in P_{\text{independent set}}(M)$. We continue with condition 3.

Claim $z'(U') = \mathrm{r}(U')$

Proof Using $z(U) = \mathrm{r}(U)$, $x(Y) = \mathrm{r}(Y)$ and the submodularity of the rank function we get

$$z(U) + x(Y) = \mathrm{r}(U) + \mathrm{r}(Y) \geq \mathrm{r}(U \cup Y) + \mathrm{r}(U \cap Y) \geq x(U \cup Y) + z(U \cap Y). \tag{6}$$

The last inequality follows since $x \in P_{\text{independent set}}(M)$, so $x(U \cup Y) \leq \mathrm{r}(U \cup Y)$ and because $U \cap Y \subseteq U$, so $z(U \cap Y) \leq \mathrm{r}(U \cap Y)$ by condition 5 (which is fulfilled by z). Comparing the two ends of (6) we get $z(U) + x(Y) = z(U - Y) + z(U \cap Y) + x(Y)$ and $x(U \cup Y) + z(U \cap Y) = x(U - Y) + z(U \cap Y) + x(Y)$. However, $z(U - Y) = x(U - Y)$ follows from (5). This implies that all inequalities in (6) are fulfilled with equation. In particular, $x(U \cup Y) = \mathrm{r}(U \cup Y)$, which proves the claim by the definition of z' and U'. □

This claim immediately implies condition 4:

$$z'(S) = z'(U') + z'(S - U') = \mathrm{r}(U') + \mu' \cdot p(S - U') + q(S - U') = \mathrm{r}(S) \qquad (7)$$

by the definitions of z' and μ'. Next we show the following.

Claim $\mu < \mu'$ and $U \neq U' \neq S$.

Proof Since $U \subseteq U'$ and because conditions 2 and 4 are fulfilled by z we get

$$r(S) = z(S) = z(U') + z(S - U') = z(U') + \mu \cdot p(S - U') + q(S - U').$$

Comparing this with (7) we get

$$(\mu' - \mu) \cdot p(S - U') = z(U') - \mathrm{r}(U') = z(U') - z'(U').$$

We know that $x \leq z$ and $x \neq z$ (since the algorithm did not terminate in *Step 1*). This together with (5) implies $x(t) < z(t)$ for some $t \in Y \subseteq U'$. Since $z'(s) = x(s) \leq z(s)$ for all $s \in U'$, we get $z'(U') < z(U')$. Hence $(\mu' - \mu) \cdot p(S - U') > 0$ which proves $\mu < \mu'$ and $U' \neq S$ by the positivity of p. Finally, $\mathrm{r}(U') = z'(U')$ was proved above, so now we have $\mathrm{r}(U') < z(U')$. Since condition 5 was met by z, this implies $U' \not\subseteq U$. □

Last, we show that the algorithm maintains condition 1. $z(s)' \leq \mu' \cdot p(s) + q(s)$ is clear by the definition of z' if $s \in S - U'$. If, on the other hand, $s \in U'$ then by the above claim and the positivity of p we have

$$z'(s) = x(s) \leq z(s) \leq \mu \cdot p(s) + q(s) < \mu' \cdot p(s) + q(s).$$

Similarly,

$$z'(s) = \mu' \cdot p(s) + q(s) > \mu \cdot p(s) + q(s) = z(s) \geq 0$$

is clear if $s \in S - U' \subseteq S - U$. And in the $s \in U'$ case $z'(s) = x(s) \geq 0$ by $x \in P_{\text{independent set}}(M)$.

We have shown that the algorithm keeps maintaining all conditions 1–5 while it also keeps strictly increasing U (by $U \subseteq U'$, $U \neq U'$). Therefore it terminates after at most $|S|$ iterations and provides a μ, U and z such that these fulfill conditions 1–5 and $z \in P_{\text{base}}(M)$. (The latter is true since $z \in P_{\text{independent set}}(M)$ and $z(S) = \mathrm{r}(S)$ follow from *Step 1* of the algorithm and condition 4, respectively, which together imply $z \in P_{\text{base}}(M)$ by Theorem 1.) Hence all optimality conditions are met at termination.

To complete the proof we show how to initialize the process. Let

$$\mu_0 := \max_{s \in S} \frac{\mathrm{r}(S) - \mathrm{r}(S - \{s\}) - q(s)}{p(s)}$$

and let s_0 be an element on which this maximum is attained. Furthermore, let $z_0 := \mu_0 \cdot p + q$. Then $z_0(s) \geq \mathrm{r}(S) - \mathrm{r}(S - \{s\})$ for all $s \in S$ and $z_0(s_0) = \mathrm{r}(S) - \mathrm{r}(S - \{s_0\})$. These imply $z_0 \geq 0$ and that $z_0(s_0) = 1$ if s_0 is a bridge (an element common to every base) and $z_0(s_0) = 0$ otherwise. Run Cunningham's algorithm (Theorem 4) on z_0 and assume it gives $x_0 \in P_{\text{independent set}}(M)$ and $U_0 \subseteq S$ such that $x_0 \leq z_0$, $x_0(U_0) = \mathrm{r}(U_0)$ and $x_0(s) = z_0(s)$ for all $s \in S - U_0$.

First assume $x_0(S) = \mathrm{r}(S)$. Then we claim that $x := x_0$, $\mu := \mu_0$ and $U := S - s_0$ satisfy all optimality criteria (see the first paragraph of the proof), therefore the process can immediately stop and output these. To show this, first observe that $x_0(S) = \mathrm{r}(S)$ implies $x_0 \in P_{\text{base}}(M)$. $x_0 \leq \mu_0 \cdot p + q$ is evident by $x_0 \leq z_0$. So we need to show $x_0(U) = \mathrm{r}(U)$ and $x_0(s_0) = z_0(s_0)$ (since s_0 is the single element of $S - U$). For this, first assume that s_0 is not a bridge. Then $z_0(s_0) = 0$, so $x_0(s_0) = 0$ by $x_0 \leq z_0$. Furthermore,

$$x_0(U) = x_0(S) - 0 = \mathrm{r}(S) = \mathrm{r}(U)$$

since s_0 is not a bridge. Now assume that s_0 is a bridge. Then $z_0(s_0) = 1$. Since $x_0 \in P_{\text{base}}(M)$, $x_0(s_0) = 1$ must also hold (because s_0 is present in every base so $x_0(s_0)$ is a convex combination of 1's). Finally,

$$x_0(U) = x_0(S) - 1 = \mathrm{r}(S) - 1 = \mathrm{r}(U).$$

Now assume $x_0(S) < \mathrm{r}(S)$. Then define μ' and z' as in *Step 2.* of the algorithm with U_0 instead of U' and x_0 instead of x. We claim that μ', z' and U_0 fulfill conditions 1–5 above thus they correctly initialize the algorithm. This is again evident for conditions 2 and 5. Condition 3 comes from $z'(U_0) = x_0(U_0) = \mathrm{r}(U_0)$. From this, condition 4, $z'(S) = \mathrm{r}(S)$ is shown in the same way as in (7) above (substituting $U' = U_0$). Furthermore, comparing (7) with

$$\mathrm{r}(S) > x_0(S) = x_0(U_0) + x_0(S - U_0) = \mathrm{r}(U_0) + \mu_0 \cdot p(S - U_0) + q(S - U_0)$$

(where we used that $x_0(s) = z_0(s)$ for all $s \in S - U_0$) gives us $\mu_0 < \mu'$. From this, condition 1 can be shown exactly the same way as above. □

To obtain a running time analysis of *Algorithm 1* we first mention that the running time of every iteration is dominated by invoking Cunningham's algorithm. However, analyzing the running time of Cunningham's algorithm is not that straightforward: in its original form we get the not too appealing bound that it takes at most $|S|^9$ iterations, each of which consists of at most $|S|^6$ independence oracle calls and $\mathcal{O}(|S|^2)$ further elementary operations. The only advantage of this form of the algorithm is that it only performs additive arithmetic on the input data. However, a variant of Cunningham's algorithm is also mentioned in [3]: by applying Gaussian elimination in each iteration, the number of iterations can be reduced to $|S|^6$ such that each iteration consists of at most $|S|^3$ independence oracle calls and $\mathcal{O}(|S|^3)$ further elementary operations. Since *Algorithm 1* uses non-additive arithmetics on the input data anyways, it is obviously better to use this modified version of Cunningham's algorithm for the purposes of *Algorithm 1*. Since it was shown above that it terminates after at most $|S|$ iterations, we get that running *Algorithm 1* takes at most $|S|^{10}$ independence oracle calls and $\mathcal{O}(|S|^{10})$ further elementary operations on the input data. This is obviously still far from being appealing; however, it should be noted that substantially better running times can probably be achieved in specific applications. Indeed, a single running of an independence oracle typically reveals much more about the underlying matroid than just the question of independence of a certain subset and the extra information could be sufficient to reduce the necessary number of oracle calls in each iteration of Cunningham's algorithm. Furthermore, since *Algorithm 1* uses Cunningham's algorithm as a “black box”, it could be replaced by any other method with a better running time that solves the problem of Theorem 4 for a certain, special class of matroids.

We remark that in the minimax relation of Theorem 5 and the algorithm of the proof of Theorem 7 we assumed the strict positivity of p (as it was motivated by the Matroid Base Game). However, this could easily be relaxed to $p \geq 0$ in both cases. Then no μ corresponding to Theorem 5 may exist: if $N = \{s \in S : p(s) = 0\}$ then an appropriate μ exists if and only if $q(S-U) \geq \mathrm{r}(S) - \mathrm{r}(U)$ holds for all subsets U for which $U \cup N = S$ (as it can be read from the proof of Theorem 5). If that condition is assumed then the maximum in Theorem 5 should be taken across subsets $U \cup N \neq S$. The algorithm in the proof of Theorem 7 works with no modification, as it also maintains the condition $U \cup N \neq S$. The only change in the proof of Theorem 7 would be that the maximum in the initialization step is taken across $s \notin N$ and if the above necessary and sufficient condition on the existence of μ does not hold then that is also easily revealed during initializaton.

4 The Common Base Game

The following generalization of Definition 1 is fairly natural.

Definition 2 Assume that the matroids $M_1 = (S, \mathscr{I}_1)$ and $M_2 = (S, \mathscr{I}_2)$ are given that have a common base. Assume further that $c, d \in \mathbb{R}^S$, $d > 0$ are also given. In the *Common Base Game* the Attacker chooses an element $s \in S$ and (simultaneously) the Defender chooses a common base B of M_1 and M_2. Then the payoff paid by the Defender to the Attacker is $d(s) - c(s)$ or $-c(s)$, if $s \in B$ or $s \notin B$, respectively.

In this section we generalize some of the results on the Matroid Base Game to the Common Base Game. This is made possible by the fact that the generalizations of the results of Sect. 2 on matroid polyhedra exist for the intersection of two matroids too. Denote by $P_{\text{common independent set}}(M_1, M_2)$ and $P_{\text{common base}}(M_1, M_2)$ the convex hulls of incidence vectors of common independent sets and common bases of M_1 and M_2, respectively. (Whenever we mention $P_{\text{common base}}(M_1, M_2)$ below, we assume that it is non-empty, that is, M_1 and M_2 have a base in common.) The following fundamental result is again due to Edmonds [13, Corollary 41.12b].

Theorem 8

$$P_{common\ independent\ set}(M_1, M_2) = P_{independent\ set}(M_1) \cap P_{independent\ set}(M_2).$$

Denote the rank functions of M_1 and M_2 by r_1 and r_2, respectively. Furthermore, let $\bar{\mathrm{r}}(U) = \min\{\mathrm{r}_1(Y) + \mathrm{r}_2(U-Y) : Y \subseteq U\}$ for all $U \subseteq S$. Then by Edmonds' classic matroid intersection theorem [13, Theorem 41.1] $\bar{\mathrm{r}}(U)$ is the maximum size of a common independent set of M_1 and M_2 contained in U. The following descriptions of $P_{\text{common independent set}}(M_1, M_2)$ and $P_{\text{common base}}(M_1, M_2)$ follow easily from Theorem 8.

Theorem 9

$$P_{\textit{common independent set}}(M_1,M_2) = \{x \in \mathbb{R}^S : x(U) \le \bar{\mathrm{r}}(U) \textit{ for all } U \subseteq S,\\ x(s) \ge 0 \textit{ for all } s \in S\}$$

and

$$P_{\textit{common base}}(M_1,M_2) = \{x \in \mathbb{R}^S : x(U) \le \bar{\mathrm{r}}(U) \textit{ for all } U \subseteq S,\\ x(S) = \bar{\mathrm{r}}(S),\\ x(s) \ge 0 \textit{ for all } s \in S\}\,.$$

Furthermore, the following generalization of Theorem 2 is also true (although it is a much deeper result than Theorem 2):

Theorem 10 [13, Corollary 41.12h] *Let $M_1 = (S,\mathscr{I}_1)$ and $M_2 = (S,\mathscr{I}_2)$ be matroids and $z \in \mathbb{R}^S$, $z \ge 0$ an arbitrary non-negative vector. Then*

$$\max\{x(S) : x \in P_{\textit{common independent set}}(M), x \le z\} =\\ \min\{\bar{\mathrm{r}}(U) + z(S-U) : U \subseteq S\}.$$

This, in turn, yields a description of the up-hull of $P_{\text{common base}}(M_1,M_2)$ the same way as Theorem 2 implied Corollary 3.

Corollary 11 [13, Section 41.4b]

$$P^{\uparrow}_{\textit{common base}}(M_1,M_2) = \{x \in \mathbb{R}^S : x(S-U) \ge \bar{\mathrm{r}}(S) - \bar{\mathrm{r}}(U) \textit{ for all } U \subseteq S\}\,.$$

Finally, the extension of Cunningham's algorithm of Theorem 4 corresponding to Theorem 10 was also given by Cunningham.

Theorem 12 [3] *Assume that the matroids $M_1 = (S,\mathscr{I}_1)$ and $M_2 = (S,\mathscr{I}_2)$ are given by independence testing oracles and a vector $z \in \mathbb{Q}^S$, $z \ge 0$ is also given. Then there exists a strongly polynomial algorithm that computes a vector $x \in P_{\textit{common independent set}}(M_1,M_2)$, $x \le z$ and a subset $U \subseteq S$ such that $x(U) = \bar{\mathrm{r}}(U)$ and $x(s) = z(s)$ for all $s \in S-U$ hold.*

Cunningham's algorithm in its original form did not compute a decomposition of x as a convex combination of incidence vectors of common independent sets, but that hiatus is filled by [13, Theorem 41.13].

Based on all these results, the generalization of Theorem 5 also follows.

Theorem 13 *Assume that the matroids $M_1 = (S,\mathscr{I}_1)$ and $M_2 = (S,\mathscr{I}_2)$ have a common base and let the vectors $d, c \in \mathbb{R}^S$, $d > 0$ be given. Then the Nash-equilibrium payoff of the Common Base Game is equal to*

$$\min\{\mu : \mu \cdot p + q \in P^{\uparrow}_{\textit{common base}}(M_1,M_2)\} = \max_{U \subseteq S, U \ne S} \frac{\bar{\mathrm{r}}(S) - \bar{\mathrm{r}}(U) - q(S-U)}{p(S-U)},$$

where $p(s) = \frac{1}{d(s)}$ and $q(s) = \frac{c(s)}{d(s)}$ for all $s \in S$.

Proof All steps of the proof of Theorem 5 extend straightforwardly to the Common Base Game. In particular, the Defender's task is equivalent to

$$\min\{\mu : \mu \cdot p + q \in P^{\uparrow}_{\text{common base}}(M_1, M_2)\}.$$

From this, the minimax relation of the theorem follows the same way from Corollary 11 as did Theorem 5 from Corollary 3. □

We omit formulating the straightforward generalization of Proposition 6 which is also true with an identical proof.

4.1 Algorithmic results on the Common Base Game

Unfortunately, the proof of Theorem 7 does not generalize to the common base case since it relies on the submodularity of the rank function which is not true for $\bar{\mathrm{r}}$. On the other hand, the polynomial time solvability of the game is immediate from the above results by binary search.

Corollary 14 *Assume that the matroids $M_1 = (S, \mathscr{I}_1)$ and $M_2 = (S, \mathscr{I}_2)$ have a common base and they are given by independence testing oracles. Assume further that the vectors $d, c \in \mathbb{Q}^S$, $d > 0$ are also given. Then the Nash-equilibrium payoff of the Common Base Game and optimum mixed strategies for both players can be computed in polynomial time.*

Proof Testing whether the Nash-equilibrium payoff of the game is (strictly) bigger than a given value ν amounts to testing whether the common optimum in Theorem 13 is bigger than ν. This is equivalent to $z := \nu \cdot p + q \notin P^{\uparrow}_{\text{common base}}(M_1, M_2)$. This, in turn, can be decided with Cunningham's algorithm from Theorem 12: $z \notin P^{\uparrow}_{\text{common base}}(M_1, M_2)$ if and only if $x(S) < \bar{\mathrm{r}}(S)$ holds for the vector x computed by the algorithm. This proves the corollary by performing binary search on ν: after finding the common optimum of Theorem 12 one can compute optimum mixed strategies for both players analogously to what is written in the first paragraph of the proof of Theorem 7. □

Furthermore, we describe an algorithm below that is strongly polynomial in certain relevant special cases. The following algorithm is simpler than Algorithm 1 and it is an adaptation of the one in the proof of [13, Theorem 42.7]. The algorithm can be initialized with any subset $U \subseteq S$, $U \neq S$ and then it keeps iterating the following steps (with p and q being the same as above).

Algorithm 2.

Step 1. Let

$$\mu := \frac{\bar{\mathrm{r}}(S) - \bar{\mathrm{r}}(U) - q(S-U)}{p(S-U)}$$

Step 2. Run Cunningham's algorithm of Theorem 12 on $z := \mu \cdot p + q$. Assume it gives $x \in P_{\text{common independent set}}(M_1, M_2)$ and $U' \subseteq S$ such that $x \leq z$, $x(U') = \bar{\mathrm{r}}(U')$ and $x(s) = z(s)$ for all $s \in S - U'$.

Step 3. If $x(S) = \bar{\mathrm{r}}(S)$ then STOP and output μ and U'. Otherwise continue at *Step 1* with U' instead of U.

$x(S) < \bar{\mathrm{r}}(S)$ and $x(U') = \bar{\mathrm{r}}(U')$ show $U' \neq S$ as long as the algorithm does not terminate (implying that the definition of μ in *Step 1* is valid). If the algorithm terminates then it finds the optima of Theorem 13 (and thus it solves the Common Base Game). Indeed,

$$\bar{\mathrm{r}}(S) = x(S) = x(U') + x(S - U') = \bar{\mathrm{r}}(U') + \mu \cdot p(S - U') + q(S - U')$$

shows $\mu = \frac{\bar{\mathrm{r}}(S) - \bar{\mathrm{r}}(U') - q(S - U')}{p(S - U')}$ and $x(S) = \bar{\mathrm{r}}(S)$ implies $x \in P_{\text{common base}}(M)$ and hence $z = \mu \cdot p + q \in P^{\uparrow}_{\text{common base}}(M)$, so μ is the common optimum value in Theorem 13. Although we cannot prove (in general) that the algorithm terminates in strongly polynomial time, the following proposition at least implies that it is finite.

Proposition 15 *The value of μ keeps strictly increasing during Algorithm 2.*

Proof Assume that the process did not terminate in *Step 3* with μ, U' and x (meaning that $x(S) < \bar{\mathrm{r}}(S)$) and denote by μ' the value computed in *Step 1* of the next iteration. Then from the definition of μ' we have

$$\bar{\mathrm{r}}(S) = \bar{\mathrm{r}}(U') + \mu' \cdot p(S - U') + q(S - U')$$

and $x(S) < \bar{\mathrm{r}}(S)$ implies

$$\bar{\mathrm{r}}(S) > x(S) = x(U') + x(S - U') = \bar{\mathrm{r}}(U') + \mu \cdot p(S - U') + q(S - U').$$

Comparing the two proves $\mu < \mu'$ as claimed. □

Since the subset U determines the value of μ, the above proposition indeed shows that the process is finite.

Note that in the following proposition the notation $\big|\{p(U) : U \subseteq S\}\big|$ refers to the number of different values the sum $\sum_{s \in U} p(s)$ can attain if U ranges over all subsets $U \subseteq S$ (and analogously for q).

Proposition 16 *Algorithm 2 is strongly polynomial if either $\big|\{p(U) : U \subseteq S\}\big|$ or $\big|\{q(U) : U \subseteq S\}\big|$ is bounded from above by a fixed polynomial of $|S|$.*

Proof Denote $f(U) = \bar{\mathrm{r}}(U) + q(S - U)$ for all $U \subseteq S$. We will prove that, except for the very last iteration, the value of $p(U)$ keeps strictly increasing during the process and the running of the algorithm can be separated into two phases (each possibly empty) such that the value of $f(U)$ keeps strictly decreasing during the first one and it keeps strictly increasing during the second. Since $\big|\{f(U) : U \subseteq S\}\big| \leq |S| \cdot \big|\{q(U) : U \subseteq S\}\big|$ is obvious, this will prove both statements of the proposition (as it will imply that Algorithm 2 must terminate after at most $\min\big\{|\{p(U) : U \subseteq S\}|, 2 \cdot |\{f(U) : U \subseteq S\}|\big\}$ iterations).

Denote by μ, x, U' and μ', x', U'' the values of the corresponding parameters generated in two consecutive iterations of the algorithm and assume that the process did not terminate even in the second one.

The properties of x' written in *Step 2* imply $x'(S) = x(U'') + x(S - U'') = f(U'') + \mu' \cdot p(S - U'')$. Furthermore, $x'(S) < \bar{\mathrm{r}}(S)$ since the algorithm did not terminate with U''. These, together with the definition of μ' give

$$f(U'') + \mu' \cdot p(S - U'') < f(U') + \mu' \cdot p(S - U'). \tag{8}$$

Next, we have $x(S) = f(U') + \mu \cdot p(S - U')$ implied by properties of x written in *Step 2*. Furthermore, $x \in P_{\text{common base}}(M_1, M_2)$ and $x \leq \mu \cdot p + q$ give $x(S) = x(U'') + x(S - U'') \leq f(U'') + \mu \cdot p(S - U'')$. Together these yield

$$f(U') + \mu \cdot p(S - U') \leq f(U'') + \mu \cdot p(S - U''). \tag{9}$$

From (8) and (9) we get

$$\mu \cdot (p(S - U') - p(S - U'')) \leq f(U'') - f(U') < \mu' \cdot (p(S - U') - p(S - U'')). \tag{10}$$

Comparing the ends of (10) and using $\mu < \mu'$ from Proposition 15 we get $p(S - U') - p(S - U'') > 0$ and thus $p(U') < p(U'')$ as claimed. This, together with (10) implies $f(U'') - f(U') < 0$ if $\mu' \leq 0$ and $f(U'') - f(U') > 0$ if $\mu > 0$. This proves what was claimed on the behaviour of $f(U)$ above by Proposition 15. □

Corollary 17 *The Common Base Game is solvable in strongly polynomial time if $c(s) = 0$ for all $s \in S$ (and d is arbitrary) or if $d(s) = 1$ for all $s \in S$ (and c is arbitrary).*

Proof Immediately from Proposition 16.

Although Proposition 16 may guarantee the strongly polynomial time solvability of the game in other special cases raised by applications, it remains open if the same holds in general.

4.2 Directed strength and the Arborescence Game

To conclude this paper, we briefly discuss an application of the results of this section: a directed version of the notion of graph strength (see Eq.(1)) and the corresponding Spanning Tree Game mentioned in Sect. 1. (In [2, Section 6] a notion distantly related to strength was considered on digraphs. The following version, which seems to be new, is different from that and it is a direct analogue of the undirected notion.)

Assume a digraph $D = (V, S)$ with vertex set V and arc set S is given. Call a subset of the nodes $R \subseteq V$ a *source set* if every node of D is reachable from a node in R via a directed path. A vertex $r \in V$ is a *source node* if $\{r\}$ is a single-element source set. Assume that D has a source node. For every arc set $U \subseteq S$, denote by $\mathrm{source}(D - U)$ the minimum cardinality of a source set in the digraph obtained from D by deleting U. (In other words, $\mathrm{source}(D - U)$ is the number of weak components in a maximum size branching of $D - U$.) Assume further that a positive weight function $p : S \to \mathbb{R}^+$ is given on the arcs. Then we can define the *directed strength* $\overrightarrow{\sigma}_p(D)$ in the following way:

$$\overrightarrow{\sigma}_p(D) = \min\left\{ \frac{p(U)}{\mathrm{source}(D - U) - 1} : U \subseteq S, \mathrm{source}(D - U) > 1 \right\}.$$

Recall that an *arborescence* of D is a subset A of the arcs that is a spanning tree of the underlying undirected graph such that the digraph (V,A) has a source node. (It is well-known and elementary that the existence of an arborescence is equivalent to the existence of a source node.) Define the *Arborescence Game*, the straightforward analogue of the Spanning Tree Game in the following way: given the cost function $c \in \mathbb{R}^S$ and the weight function $d \in \mathbb{R}^S$, $d > 0$, the Attacker chooses an arc $s \in S$, the Defender chooses an arborescence $A \subseteq S$ and then the payoff paid by the Defender to the Attacker is $d(s) - c(s)$ or $-c(s)$, if $s \in A$ or $s \notin A$, respectively.

Theorem 18 *The Nash-equilibrium payoff of the Arborescence Game is*

$$\max_{U \subseteq S, U \neq \emptyset} \frac{\text{source}(D-U) - 1 - q(U)}{p(U)},$$

where $p(s) = \frac{1}{d(s)}$ *and* $q(s) = \frac{c(s)}{d(s)}$ *for all* $s \in S$.

Proof It is well-known that arborescences can be described as the common bases of two matroids (see [13, Section 52.10], for instance): let $M_1 = (S, \mathscr{I}_1)$ be the cycle matroid of the underlying undirected graph of D and let $M_2 = (S, \mathscr{I}_2)$ be the partition matroid in which $U \in \mathscr{I}_2$ for an arc set U if no two arcs in U enter a common vertex. One easily checks that the common bases of M_1 and M_2 are the arborescences of D and $\bar{\mathrm{r}}(S) - \bar{\mathrm{r}}(S-U) = \text{source}(D-U) - 1$ for all subsets $U \subseteq S$ (where $\bar{\mathrm{r}}(U)$ is the maximum size of a common independent set contained in U). Then the theorem follows immediately from Theorem 13. □

We remark that an optimum mixed strategy for the Attacker can be determined from any subset U at which the maximum in Theorem 18 is attained by applying the formula of Proposition 6 on $S-U$ (see the remark after Theorem 13). The following theorem shows the relation of the game to $\vec{\sigma}_p(D)$.

Theorem 19 *Assume that a digraph* $D = (V,S)$ *is given that has a source node and a weight function* $p \in \mathbb{R}^S$, $p > 0$ *is also given. Then* $\vec{\sigma}_p(D)$ *is the reciprocal of the Nash-equilibrium payoff of the Arborescence Game with* $d(s) = \frac{1}{p(s)}$ *and* $c(s) = 0$ *for all* $s \in S$. *Furthermore,* $\vec{\sigma}_p(D)$ *as well as optimum mixed strategies for both players can be computed in strongly polynomial time.*

Proof Immediately from Corollary 17 and the proof of Theorem 18. □

References

1. T. Alpcan and T. Başar, Network security: A decision and game-theoretic approach, Cambridge University Press, New York (2010)
2. W. H. Cunningham, Optimal attack and reinforcement of a network, Journal of the ACM (JACM), vol. 32(3), pp. 549-561 (1985).
3. W. H. Cunningham, Testing membership in matroid polyhedra, Journal of Combinatorial Theory, Series B, vol. 36(2), pp 161-188 (1984).
4. A. Gueye, J. C. Walrand and V. Anantharam, Design of network topology in an adversarial environment, Proc. of the 1st International Conference on Decision and Game Theory for Security, GameSec10, Berlin, Germany, pp. 1-20 (2010).

5. A. Gueye, J. C. Walrand and V. Anantharam, A network topology design game: How to choose communication links in an adversarial environment, Proc. of the 2nd International ICST Conference on Game Theory for Networks, GameNets, vol. 11 (2011).
6. D Gusfield, Connectivity and edge-disjoint spanning trees, Information Processing Letters, vol. 16.2, pp. 87-89 (1983)
7. Z. Han, Game theory in wireless and communication networks: theory, models, and applications, Cambridge University Press, New York (2012)
8. A. Laszka and D. Szeszlér, Hide and Seek in Digital Communication: The Steganography Game, Proc. 9th Hungarian-Japanese Symposium on Discrete Mathematics and Its Applications, Fukuoka, Japan, pp. 126-136 (2015)
9. R. Machado and S. Tekinay, A survey of game-theoretic approaches in wireless sensor networks, Computer Networks, vol. 52(16), pp 3047-3061 (2008)
10. M. H. Manshaei, Q. Zhu, T. Alpcan, T. Bacsar and J. P. Hubaux, Game theory meets network security and privacy, ACM Computing Surveys (CSUR), vol. 45.3 , Article No. 25 (2013)
11. J. Matoušek, B. Gärtner, Understanding and Using Linear Programming, 226 pages. Springer, Berlin, Heidelberg (2007)
12. J. v. Neumann, Zur Theorie der Gesellschaftsspiele, Mathematische Annalen, vol. 100(1), pp. 295-320 (1928)
13. A. Schrijver, Combinatorial Optimization: Polyhedra and Efficiency, Algorithms and Combinatorics vol. 24, 1879 pages. Springer, Berlin, Heidelberg (2003)
14. M. Tambe, Security and game theory: algorithms, deployed systems, lessons learned. Cambridge University Press, New York (2012)