Spanning-Tree Games

Dan Hefetz[1]
Department of Computer Science, Ariel University, Israel

Orna Kupferman[2]
School of Computer Science and Engineering, The Hebrew University, Israel

Amir Lellouche
Department of Computer Science, Weizmann Institute of Science, Israel

Gal Vardi
School of Computer Science and Engineering, The Hebrew University, Israel

Abstract

We introduce and study a game variant of the classical spanning-tree problem. Our *spanning-tree game* is played between two players, MIN and MAX, who alternate turns in jointly constructing a spanning tree of a given connected weighted graph G. Starting with the empty graph, in each turn a player chooses an edge that does not close a cycle in the forest that has been generated so far and adds it to that forest. The game ends when the chosen edges form a spanning tree in G. The goal of MIN is to minimize the weight of the resulting spanning tree and the goal of MAX is to maximize it. A strategy for a player is a function that maps each forest in G to an edge that is not yet in the forest and does not close a cycle.

We show that while in the classical setting a greedy approach is optimal, the game setting is more complicated: greedy strategies, namely ones that choose in each turn the lightest (MIN) or heaviest (MAX) legal edge, are not necessarily optimal, and calculating their values is NP-hard. We study the approximation ratio of greedy strategies. We show that while a greedy strategy for MIN guarantees nothing, the performance of a greedy strategy for MAX is satisfactory: it guarantees that the weight of the generated spanning tree is at least $\frac{w(MST(G))}{2}$, where $w(MST(G))$ is the weight of a maximum spanning tree in G, and its approximation ratio with respect to an optimal strategy for MAX is $1.5 + \frac{1}{w(MST(G))}$, assuming weights in $[0, 1]$. We also show that these bounds are tight. Moreover, in a stochastic setting, where weights for the complete graph K_n are chosen at random from $[0, 1]$, the expected performance of greedy strategies is asymptotically optimal. Finally, we study some variants of the game and study an extension of our results to games on general matroids.

2012 ACM Subject Classification Mathematics of computing → Graph algorithms

Keywords and phrases Algorithms, Games, Minimum/maximum spanning tree, Greedy algorithms

Digital Object Identifier 10.4230/LIPIcs.MFCS.2018.35

Related Version A full version of the paper is available at `http://www.cs.huji.ac.il/~ornak/publications/mfcs18a.pdf`.

Acknowledgements We thank Yuval Peled for helpful discussions.

[1] The research leading to this paper was done when the author was visiting the Hebrew University.
[2] The research leading to this paper has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013).

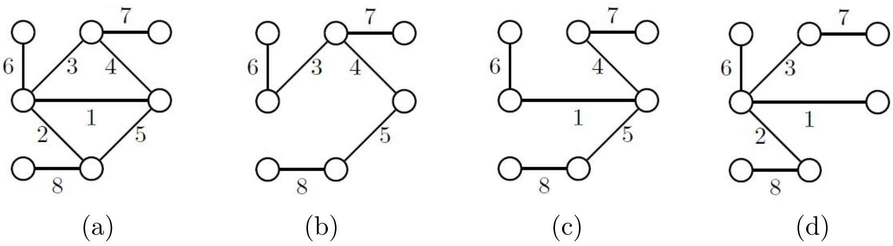

Figure 1 A weighted graph (a), its maximal spanning tree (b), and the outcomes of an optimal strategy (c) and a greedy one (d).

1 Introduction

The fundamental *minimum (respectively, maximum) spanning tree problem* receives as an input a connected edge-weighted undirected graph and searches for a spanning tree, namely an acyclic subgraph that connects all vertices, with a minimum (respectively, maximum) weight. The problem can be solved efficiently [19, 26]. It has attracted much attention, has led to a lot of research on algorithms, and has many applications [28, 10, 14].

We introduce and study a natural game variant of the classical problem. Our *spanning-tree game* is played between two players, MIN and MAX, who alternate turns in jointly constructing a spanning tree of a given connected weighted graph $G = \langle V, E, w \rangle$. Starting with the empty graph, in each turn a player chooses an edge that does not close a cycle in the forest that has been generated so far and adds it to that forest. The game ends when the chosen edges form a spanning tree in G, that is, after $|V| - 1$ turns. The goal of MIN is to minimize the weight of the resulting spanning tree and the goal of MAX is to maximize it. A *strategy* for a player is a function that maps each forest in G to one of its legal moves, namely, it maps a forest $F \subseteq E$ to an edge $e \in E \setminus F$ such that $F \cup \{e\}$ is also a forest. Given two strategies π_{max} and π_{min}, we define the *outcome* of π_{max} and π_{min} as the spanning tree obtained when MAX and MIN follow π_{max} and π_{min}, respectively, in a turn-based game in which MAX moves first. The *value* of a strategy π_{max} of MAX is the minimum over all strategies π'_{min} of MIN of the weight of the spanning tree that is the outcome of the game in which MAX follows π_{max} and MIN follows π'_{min}. Then, an optimal strategy for MAX is a strategy with a maximum value. Thus, an optimal strategy for MAX is one that obtains the maximal value against the most hostile behavior (intuitively, the "most minimizing" strategy) of MIN. The value of a strategy for MIN is defined dually. In particular, an optimal strategy for MIN is one that obtains the minimal value against the "most maximizing" strategy for MAX. In this paper we focus on values of strategies of MAX. Indeed, unless we bound the ratio between the weights of the heaviest and lightest edges in the graph, we cannot bound the "damage" that MAX can cause MIN, namely the ratio between the performance of min strategies and the minimum spanning tree, making the study of the game setting from the viewpoint of MIN less interesting.

▶ **Example 1.** Consider the weighted graph G appearing in Figure 1 (a). The weight of G's (unique, in this example) maximum spanning tree is 33 (see (b)). An optimal strategy for MAX chooses in its first two moves the edges with weights 5 and 4, leading, against an optimal strategy of MIN, to the spanning tree of weight 31 appearing in (c).

The transition from the classical *one-player* setting of the spanning-tree problem to a *two-player* setting corresponds to a transition from *closed systems*, which are completely under our control, to *open systems*, in which we have to contend with adversarial environments. Such a transition has been studied in computer science in logic [8, 27], complexity [6], and temporal reasoning [23], and it attracts growing attention now in *algorithmic game theory*, cf. [24]. Our work here studies this transition in graph theory. For the basic problem of reachability, the two-player setting has given rise to *alternating graph reachability* [8]. We find it very interesting to study how other basic problems and concepts in graph algorithms evolve when we shift to a two-player setting [20]. Several graph games of this type were previously studied. For example, consider the general setting in which MAX and MIN alternately claim edges of a graph G while making sure the graph they build together satisfies some monotone decreasing property. The *Turán numbers* and *Saturation numbers* refer to the number of edges that can be claimed while the property is maintained [13, 17]. Likewise, researchers have studied the *game chromatic number* of G, namely the smallest k for which MIN has a strategy to color all vertices in a game in which MAX and MIN alternately properly color the vertices of G using the colors $\{1, \ldots, k\}$ [1]. Finally, a game variant of the *maximum-flow problem*, where the algorithm can direct the flow only in a subset of the vertices is studied in [21].

Before we continue to describe our results, let us survey several games that have been studied and are based on minimum or maximum spanning trees. In the *cooperative minimum cost spanning tree game* [7, 2], the cost allocation between users of a minimum spanning tree is considered. Different properties of this cooperative game have been studied, such as the core and the nucleolus [15, 16], the Shapley value [18], and more [11]. The *Stackelberg minimum spanning tree game* [4, 5] is a one-round two-player network pricing game. The game is played on a graph, whose edges are colored either red or blue, with the red edges having a given fixed cost. The first player chooses an assignment of prices to the blue edges, and the second player then buys the cheapest possible minimum spanning tree, using any combination of red and blue edges. The goal of the first player is to maximize the total price of purchased blue edges. *Shannon's switching game* is another related two-player game. Two players take turns coloring the edges of an arbitrary graph. One player has the goal of connecting two distinguished vertices by a path of edges of her color. The other player aims to prevent this by using her color instead (or, equivalently, by erasing edges) [22, 3].

The classical maximum spanning-tree problem can be solved efficiently. Indeed, the forests embodied in a graph induce a *matroid* [25], and thus a greedy approach is optimal. Accordingly, in Kruskal's algorithm [19] for the maximum spanning-tree problem, the edges are chosen in a greedy manner, where in each step an edge with a maximum weight that does not close a cycle is added.

We study greedy strategies in the spanning-tree game. There, MAX always chooses an edge with a maximum weight that does not close a cycle. We first show that the game setting is indeed more complicated. First, greedy strategies are not necessarily optimal. For example, in the graph from Example 1, a greedy strategy for MAX chooses in its first three moves the edges with weight 8, 7, and 6, leading to the spanning tree of weight 27 appearing in Figure 1 (d). In addition, we show that given a strategy for MAX, it is NP-complete to calculate its value, and NP-hardness holds already for greedy strategies. Subsequently, we turn to study how well greedy strategies for MAX perform. We evaluate them with respect to the value of the maximum spanning tree, and with respect to the value of an optimal strategy for MAX. We analyze both the general and stochastic settings. We view our findings in both evaluations as good news. Indeed, greedy strategies for MAX ensure surprisingly tight approximations in all cases.

It is not hard to see that the value of any greedy strategy for MAX is at least half the weight of a maximum spanning tree. Indeed, the tree generated by such a strategy includes at least the heavier half of the set of edges that are chosen by a greedy algorithm in the classical setting. Much harder is the study of the approximation ratio of a greedy strategy for MAX with respect to an optimal strategy for her. We show that when the weight of the maximum spanning tree tends to infinity, the approximation ratio tends to 1.5. More formally, assuming that the weights are normalized to values in $[0, 1]$ (note that such a normalization does not affect the ratio between the values of different strategies), we show an approximation ratio of $1.5 + \frac{1}{w(MST(G))}$, where $w(MST(G))$ is the weight of a maximum spanning tree of G. We show that our results are tight: for every odd integer $n \geq 1$, there exists a weighted graph $G = \langle V, E, w\rangle$ with $w(MST(G)) = 2n$, such that the value of the greedy strategy for MAX is n, whereas the value of an optimal strategy is $\lceil \frac{n}{2} \rceil + n$. Thus, the ratio between the maximal spanning tree and the value of the greedy strategy is 2, and the ratio between the values of the optimal and the greedy strategies is $1.5 + \frac{1}{w(MST(G))}$. We also show that, unlike the case of greedy strategies of MAX, one cannot bound the approximation ratio of greedy strategies of MIN. As we elaborate in Section 7, since the set of forests that are subgraphs of a given graph form the family of independent sets of a matroid, many of our results go beyond the spanning-tree problem and apply to matroids in a game setting.

We then study the approximation ratio of greedy strategies for MAX in a stochastic setting. Namely, we study the game played on complete graphs whose edge-weights are chosen by a uniform distribution over $[0, 1]$. Building on results of [12] regarding the weight of maximum and minimum spanning trees in such randomly weighted graphs, we are able to show that, in this setting, the approximation ratio of any greedy max strategy is asymptotically almost surely (a.a.s., for brevity) 1. Thus, while in the worst case the approximation ratio is 2 with respect to a maximum spanning tree and it tends to 1.5 with respect to an optimal strategy, it is a.a.s. 1 when we choose the edge-weights uniformly at random.

Finally, we study two variants of the setting. First, a finer definition of an approximation ratio, where performance of a strategy for MAX is examined with respect to all strategies of MIN, and second, a two-turn variant of the game, where MAX first chooses a forest of size k, for a parameter k of the game, and then MIN completes the forest to a spanning tree.

2 Preliminaries

2.1 Graphs and Weighted Graphs

An undirected *graph* is a pair $G = \langle V, E\rangle$, where V is a finite set and E is a set of pairs of elements of V. We refer to the elements of V as *vertices* and to the elements of E as *edges*. A graph may contain parallel edges. A *path* in G is a sequence of vertices $v_1, v_2, \ldots, v_k$ such that $\langle v_i, v_{i+1}\rangle \in E$ for all $1 \leq i < k$. A *cycle* in G is a path $v_1, v_2, \ldots, v_k$ for which $v_1 = v_k$. A graph $G = \langle V, E\rangle$ is *connected* if for every two vertices $v, v' \in V$, there is a path between v and v' in G. A *tree* is a connected graph with no cycles. A *forest* is a graph with no cycles, namely a collection of trees. A *spanning tree of* G is a tree $\langle V, T\rangle$, for a subset $T \subseteq E$. Note that the size of a spanning tree is $n - 1$. When the set V of vertices is clear from the context, we describe trees and forests by their sets of edges only.

A *weighted graph* $G = \langle V, E, w\rangle$ augments a graph with a weight function $w : E \to \mathbb{R}^+$. We extend w to subsets of E in the expected way, i.e., $w : 2^E \to \mathbb{R}^+$ is such that for all $A \subseteq E$, we have $w(A) = \sum_{e \in A} w(e)$. In the *maximum spanning tree* problem, we are given a weighted graph G and seek a spanning tree for G of a maximum weight. Note that G may have several maximum spanning trees. By abuse of notation, we use $MST(G)$ to denote any maximum spanning tree of G.

2.2 Matroids

A finite matroid $\mathcal{M}$ is a pair $\langle E, \mathcal{I} \rangle$, where E is a finite set (called *the ground set*) and $\mathcal{I}$ is a family of subsets of E (called *the independent sets*) that satisfies the following three properties: (1) $\mathcal{I}$ is not empty, (2) *The hereditary property*: If $X \in \mathcal{I}$ and $Y \subseteq X$, then $Y \in \mathcal{I}$, and (3) *The independent set exchange property*: If X and Y are in $\mathcal{I}$ and $|X| > |Y|$, then there is an element $x \in X \setminus Y$ such that $Y \cup \{x\}$ is in $\mathcal{I}$.

For a graph $G = \langle V, E \rangle$, let $\mathcal{F}_G$ be the set of forests in G. The pair $\langle E, \mathcal{F}_G \rangle$ is a matroid and is called *the cycle matroid of* G (see, e.g., [25]).

2.3 The Spanning-Tree Game

We consider a game variant of the maximum spanning tree problem: there are two players, MAX and MIN, who alternate turns in jointly constructing a spanning tree of a given weighted graph. Starting with the empty graph, in each turn, a player chooses an edge that does not close a cycle in the forest that has been generated so far and adds it to that forest. The game ends when the chosen edges are forming a spanning tree, that is, after $n - 1$ turns. The goal of MIN is to minimize the weight of the resulting spanning tree and the goal of MAX is to maximize it. Formally, we have the following.

Let $G = \langle V, E, w \rangle$ be a weighted graph, and let $\mathcal{F}_G$ be the set of all forests $F \subseteq E$. A *configuration* in the *spanning-tree game* is a forest $F \in \mathcal{F}_G$. Let $M : \mathcal{F}_G \to 2^E$ be a function that maps a configuration F to the set of all legal moves for a player when the game is in F. Formally, $M(F) = \{e \in E \setminus F \ : \text{ the graph } \langle V, F \cup \{e\} \rangle \text{ has no cycles}\}$.

A *strategy* for a player is a function $\pi : \mathcal{F}_G \to E$ that maps each configuration to one of its legal moves. Thus, for all $F \in \mathcal{F}_G$, we have $\pi(F) \in M(F)$. If $M(F) = \emptyset$ (that is, when F is already a spanning tree), then $\pi(F)$ is undefined.[3] Given two strategies π_{max} and π_{min}, we define the *outcome* of π_{max} and π_{min}, denoted $T(\pi_{max}, \pi_{min})$, as the spanning tree obtained when MAX and MIN follow π_{max} and π_{min}, respectively, in a turn-based game in which MAX moves first. Formally, $T(\pi_{max}, \pi_{min}) = \{e_1, \ldots, e_{n-1}\}$ is such that for all $1 \leq i \leq n - 1$, the following holds.

$$e_i = \left[\begin{array}{ll} \pi_{max}(\{e_1, e_2, \ldots, e_{i-1}\}) & \text{if } i \text{ is odd,} \\ \pi_{min}(\{e_1, e_2, \ldots, e_{i-1}\}) & \text{if } i \text{ is even.} \end{array} \right.$$

We use $w(\pi_{max}, \pi_{min})$ to denote the weight of $T(\pi_{max}, \pi_{min})$. Thus, $w(\pi_{max}, \pi_{min}) = w(T(\pi_{max}, \pi_{min}))$.

We refer to a strategy for MAX as a *max strategy* and to a strategy for MIN as a *min strategy*. Note that MAX moves when the current configuration has an even number of edges, and MIN moves when the configuration has an odd number of edges. Let $\mathcal{F}_G^{even}$ and $\mathcal{F}_G^{odd}$ be the subsets of $\mathcal{F}_G$ that contain forests of even and odd sizes, respectively. Let Π_{max} and Π_{min} be the set of all possible strategies for the MAX and MIN players, respectively. By the above, Π_{max} contains strategies $\pi_{max} : \mathcal{F}_G^{even} \to E$ and Π_{min} contains strategies

[3] We could have defined π to return a special signal, say $\perp$, in this case, but we ignore it and assume that the game ends after $n - 1$ rounds, so there is no need to apply a strategy from configurations that are spanning trees.

$\pi_{min} : \mathcal{F}_G^{odd} \to E$.[4] We evaluate a max strategy π_{max} by its performance against a best (that is, most minimizing) min strategy. Formally, we define the *value* of a max strategy by

$$val_{max}(\pi_{max}) = \min\{w(\pi_{max}, \pi_{min}) : \pi_{min} \in \Pi_{min}\}.$$

Since the number of strategies is finite, the above expression always has a minimum and is thus well defined. Dually, we evaluate a min strategy π_{min} by its performance against a best (that is, most maximizing) max strategy. Formally, we define the value of a min strategy by $val_{min}(\pi_{min}) = \max\{w(\pi_{max}, \pi_{min}) : \pi_{max} \in \Pi_{max}\}$. Our study here focuses on max strategies. Essentially, our choice follows from the fact that, unlike the case of max strategies, one cannot bound the ratio between the outcome of an optimal or a greedy min strategy and the minimum spanning tree. Intuitively, it follows from the fact that the performance of strategies is strongly related to our ability to guarantee a favorable outcome even if we can control only half of the choices. Such a control guarantees that MAX can add to the spanning tree at least half of the heaviest edges in a maximum spanning tree. Such a control also guarantees that MIN can add to the spanning tree at least half of the lightest edges in a minimum spanning tree. Without, however, a bound on the ratio between the heaviest and lightest edge, such a guarantee is not of much help. In the full version, we motivate this choice further and present some results on min strategies.

The following lemma is an easy useful observation on the amount of control MAX and MIN have on the outcome of the game.

▶ **Lemma 2.** *Let $G = \langle V, E, w \rangle$ be a weighted graph and let F be a forest of G. Then,* MAX *has a strategy to ensure that the outcome includes at least $\lceil |F|/2 \rceil$ edges of F, and* MIN *has a strategy to ensure that the outcome includes at least $\lfloor |F|/2 \rfloor$ edges of F.*

Proof. We prove our claim for MIN; the proof for MAX is analogous. It suffices to show that, in each of his first $\lfloor |F|/2 \rfloor$ moves, MIN can claim an edge of F. For every $1 \le i \le \lfloor |F|/2 \rfloor$, let $e_1, \ldots, e_{2i-1}$ denote the edges claimed by both players up until MIN's i-th move. In his i-th move, MIN claims an arbitrary edge $e_{2i} \in F \setminus \{e_1, \ldots, e_{2i-1}\}$ such that $\{e_1, \ldots, e_{2i-1}, e_{2i}\}$ spans a forest. Such an edge e_{2i} exists since $|F| > 2i - 1 = |\{e_1, \ldots, e_{2i-1}\}|$ and both F and $\{e_1, \ldots, e_{2i-1}\}$ are forests of G, i.e., independent sets in its cycle matroid. ◀

2.4 Optimal and Greedy Strategies

We define the following strategies:

- An *optimal max strategy* is a strategy $\pi^*_{max} \in \Pi_{max}$ such that for every strategy $\pi_{max} \in \Pi_{max}$, we have $val_{max}(\pi^*_{max}) \ge val_{max}(\pi_{max})$. Such a strategy necessarily exists as the number of max strategies is finite.
- Similarly, $\pi^*_{min} \in \Pi_{min}$ is an *optimal min strategy*, if for every strategy $\pi_{min} \in \Pi_{min}$, we have $val_{min}(\pi^*_{min}) \le val_{min}(\pi_{min})$.
- A strategy $g_{max} \in \Pi_{max}$ is a *greedy strategy* for MAX if for every configuration $F \in \mathcal{F}_G^{even}$, it holds that $g_{max}(F)$ is a heaviest edge in $M(F)$. Formally, for every configuration $F \in \mathcal{F}_G^{even}$, we have $g_{max}(F) \in \{e \in M(F) : w(e) = \max\{w(e') : e' \in M(F)\}\}$.

[4] Formally, by our definition of a strategy, every strategy for MAX and every strategy for MIN should have a well-defined legal move for every configuration in $\mathcal{F}_G$. We have chosen to restrict the definition of such strategies only to the configurations they might actually encounter during play. For completeness, one can define them for all the remaining configurations arbitrarily or, again, by using the symbol $\perp$. Also, note that strategies are *positional*, in the sense they ignore the way in which configurations have been obtained. It is easy to see that memoryfull strategies are not stronger in the spanning-tree game.

▶ Remark. There may be several optimal and greedy strategies but, from now on, for each weighted graph G we define π^*_{min}, π^*_{max}, and g_{max} as one of the strategies that satisfy the corresponding conditions and, sometimes, write "the optimal min strategy" or "the greedy max strategy". Moreover, when evaluating the performance of a greedy strategy, we consider the worst case. That is, the value of a greedy strategy is $\min\{val_{max}(g_{max}) : g_{max} \text{ is a greedy strategy in } \Pi_{max}\}$.

2.5 On the Complexity of Evaluating Strategies for MAX

Recall that the maximum spanning-tree problem can be solved in polynomial time. A possible way of computing π^*_{min} and π^*_{max} is by solving a Minmax problem, which requires exponential time. We show here that the game setting is indeed more complex than the classical one-player setting. In fact, even evaluating the value of a symbolically given max strategy is co-NP-complete, and the co-NP lower bound holds also for greedy strategies.

▶ **Theorem 3.** *Let π_{max} be a max strategy given by a linear ordering $e_1, \ldots, e_{|E|}$ of the edges in E, where π_{max} chooses in each step the edge e_j with the minimal index j for which e_j is a legal move. Let k be an integer. Deciding whether $val_{max}(\pi_{max}) > k$ is co-NP-complete. Furthermore, it is co-NP-hard already when π_{max} is a worst greedy strategy for* MAX*, that is, a greedy strategy with the lowest value.*

Proof. First, if $val_{max}(\pi_{max}) \leq k$ then there is a polynomial witness that includes the edges that MIN chooses in each turn, such that the weight of the outcome is at most k. Hence the membership in co-NP.

We now show the lower bound. Let $G = \langle V, E\rangle$ be a graph, let $S \subseteq V$, and let k be an integer. The Steiner-tree problem, namely, deciding whether there is a tree of size at most k in G that spans S, is NP-hard. We show a reduction from the Steiner tree problem. We construct a weighted graph $G' = \langle V', E', w'\rangle$ as follows. Let u_0 be a vertex in V. The set V' is obtained from V by adding k new vertices, namely $V' = V \cup \{u_1, \ldots, u_k\}$. The set E' is obtained from E by adding the edges $\{\langle u_i, u_{i+1}\rangle : 0 \leq i < k\} \cup (S \times S)$, where parallel edges are allowed. That is, an edge $e \in S \times S$ is added even if it already appears in E. For every $e \in E$ we define $w'(e) = 0$, and for every new edge $e \in E' \setminus E$ we define $w'(e) = 1$. Let π_{max} be a max strategy in which MAX first chooses edges in $\{\langle u_i, u_{i+1}\rangle : 0 \leq i < k\}$, and when it is not possible anymore she chooses edges in $S \times S$, and when it is not possible anymore she chooses edges in E. We prove that there is a tree in G that spans S and has size at most k iff $val_{max}(\pi_{max}) \leq k$. Assume that there is a tree in G that spans S and has size at most k. We denote this tree by T. Then, while MAX chooses edges in $\{\langle u_i, u_{i+1}\rangle : 0 \leq i < k\}$, MIN can choose all the edges of T and thus ensure that MAX will not be able to choose edges in $S \times S$ later. Since the edges $\{\langle u_i, u_{i+1}\rangle : 0 \leq i < k\}$ appear in every spanning tree, the value of π_{max} is k.

Assume now that there is no tree in G that spans S and has size at most k. Thus, after all the edges in $\{\langle u_i, u_{i+1}\rangle : 0 \leq i < k\}$ are chosen, there are still edges in $S \times S$ that MAX can choose, and therefore the value of π_{max} is strictly larger than k.

Finally, note that the strategy π_{max} is a worst greedy strategy for MAX, and hence the problem is co-NP-hard already for this case. ◀

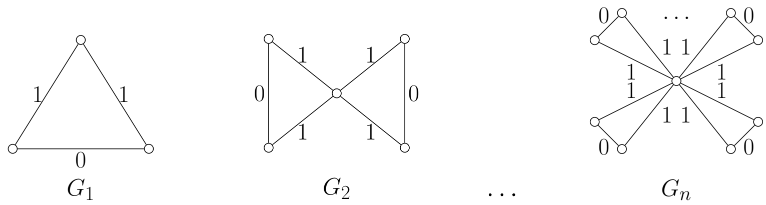

Figure 2 A sequence of weighted graphs $G_1, G_2, \ldots$ such that G_n satisfies $n = val_{max}(g_{max}) = val_{max}(\pi^*_{max}) = \frac{1}{2} \cdot w(MST(G_n))$.

3 The Performance of Optimal and Greedy Strategies w.r.t. the Maximum Spanning Tree

In the game setting, MAX has a chance to choose only half of the edges in the spanning tree. It is thus not surprising that the outcome of an optimal strategy may be only half of the weight of an MST. Below we formalize this intuition, and show that the half-ratio may be obtained already by a greedy strategy (Theorem 4) and that this upper bound is tight (Theorem 5).

▶ **Theorem 4.** *For every weighted graph G, we have that $val_{max}(g_{max}) \geq \frac{1}{2} \cdot w(MST(G))$.*

Proof. Let $G = \langle V, E, w \rangle$, and let $\langle e_1, \ldots, e_{n-1} \rangle$ be a vector of the edges of some maximum spanning tree of G, where $w(e_i) \geq w(e_{i+1})$ for every $1 \leq i < n-1$. Consider the game on G in which MAX plays according to g_{max} and MIN plays according to some strategy π_{min}. For every $1 \leq j \leq \lceil (n-1)/2 \rceil$, let x_j denote the edge of G that MAX chooses in her j-th move. For every $1 \leq j \leq \lfloor (n-1)/2 \rfloor$, let y_j denote the edge of G that MIN chooses in his j-th move. Our goal is to prove that

$$\sum_{j=1}^{\lceil (n-1)/2 \rceil} w(x_j) + \sum_{j=1}^{\lfloor (n-1)/2 \rfloor} w(y_j) \geq \frac{1}{2} \cdot \sum_{j=1}^{n-1} w(e_j).$$

We prove that, in fact, already $\sum_{j=1}^{\lceil (n-1)/2 \rceil} w(x_j) \geq \frac{1}{2} \cdot \sum_{j=1}^{n-1} w(e_j)$. Since all edge-weights are non-negative, this implies our goal.

To see this, consider an integer $0 \leq k < (n-1)/2$. Note that $|\{x_1, \ldots x_k, y_1, \ldots, y_k\}| = 2k < 2k+1 = |\{e_1, \ldots, e_{2k+1}\}|$. Since, moreover, $\{x_1, \ldots x_k, y_1, \ldots, y_k\}$ and $\{e_1, \ldots, e_{2k+1}\}$ are independent sets of a matroid (namely, the cycle matroid of G), there exists some edge $e \in \{e_1, \ldots, e_{2k+1}\} \cap M(\{x_1, \ldots x_k, y_1, \ldots, y_k\})$. Since MAX plays according to the greedy strategy, it must be that $w(x_{k+1}) \geq w(e) \geq w(e_{2k+1})$. Hence, $\sum_{j=1}^{\lceil (n-1)/2 \rceil} w(x_j) \geq \sum_{j=0}^{\lceil (n-1)/2 \rceil - 1} w(e_{2j+1}) \geq \frac{1}{2} \cdot \sum_{j=1}^{n-1} w(e_j)$, and the statement follows. ◀

▶ **Theorem 5.** *For every $n \geq 1$, there is a weighted graph G_n such that $n = val_{max}(\pi^*_{max}) = \frac{1}{2} \cdot w(MST(G_n))$. In fact, for G_n we also have $val_{max}(g_{max}) = n$.*

Proof. See the weighted graphs $G_1, G_2, \ldots$ in Figure 2. Note that $MST(G_n)$ includes all the edges with weight 1, and that MIN can ensure that all the edges with weight 0 are chosen. ◀

4 The Performance of Greedy Strategies w.r.t. Optimal Ones

In this section we study the performance of the greedy max strategy in comparison to the optimal max strategy. We first define formally what it means for two strategies to approximate each other.

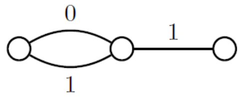

Figure 3 $val_{max}(g_{max}) = 1$ whereas $val_{max}(\pi^*_{max}) = 2$.

4.1 Approximating Strategies

Given a weighted graph $G = \langle V, E, w \rangle$, consider two max strategies $\pi_{max}, \pi'_{max} \in \Pi_{max}$ and a factor $\alpha \geq 1$. We say that π_{max} is an *α-max-approximation* of π'_{max} if

$$val_{max}(\pi_{max}) \geq 1/\alpha \cdot val_{max}(\pi'_{max}).$$

That is, intuitively, π'_{max} is at most α times better than π_{max}, where, in both cases, we assume that MIN follows an optimal min strategy.

The *max competitive ratio* of a strategy $\pi_{max} \in \Pi_{max}$ is then the smallest factor α such that π_{max} is an α-max approximation of π^*_{max}. Namely, $\frac{val_{max}(\pi^*_{max})}{val_{max}(\pi_{max})}$.

▶ Remark (Universal Approximation). We could have defined strategy approximations in a different way, by stating that π_{max} is an α-max-approximation of π'_{max} if for every strategy $\pi_{min} \in \Pi_{min}$, we have that $w(\pi_{max}, \pi_{min}) \geq \frac{1}{\alpha} \cdot w(\pi'_{max}, \pi_{min})$. We refer to such an approximation as *α-max universal approximation*. Intuitively, while in α-max-approximation the performance of the two max strategies is examined with respect to optimal (possibly different from each other) min strategies, in α-max universal approximation the performance is examined with respect to every min strategy – the same min strategy against both max strategies. In the full version, we show that α-max universal approximation is strictly finer than α-max approximation. That is, for all $\pi_{max}, \pi'_{max} \in \Pi_{max}$ and $\alpha \geq 1$, if π_{max} is an α-max universal approximation of π'_{max}, then π_{max} is an α-max approximation of π'_{max}, yet possibly π_{max} is an α-max approximation of π'_{max} but it is not an α-max universal approximation of π'_{max}. Moreover, working with a max universal approximation, the competitive ratio of the greedy strategy with respect to the optimal strategy is 2, higher than the ratio we prove in Theorem 7, when working with a max approximation.

4.2 The Competitive Ratio of Greedy Max Strategies

We turn to study the max competitive ratio of the greedy strategy. For convenience, we assume that the weight function w is normalized so that $\max\{w(e) : e \in E\} = 1$. It is easy to see that such a normalization is always possible and does not change the ratio of the weights of any two spanning trees.

▶ **Theorem 6.** *The max competitive ratio of the greedy strategy is* 2.

Proof. We first prove that g_{max} is a 2-max approximation. By Theorem 4, we have $2 \cdot val_{max}(g_{max}) \geq w(MST(G))$. In addition, as no max strategy can perform better than the weight of a maximum spanning tree, we have that $w(MST(G)) \geq val_{max}(\pi_{max})$ for all $\pi_{max} \in \Pi_{max}$. Hence, $val_{max}(g_{max}) \geq \frac{1}{2} \cdot val_{max}(\pi_{max})$ for all $\pi_{max} \in \Pi_{max}$, and we are done.

Next, in order to prove that the factor 2 is tight, consider the graph in Figure 3. It is easy to see that while an optimal max strategy would choose first the parallel edge with weight 1, leading to a spanning tree of weight 2, a greedy strategy may choose first the edge on the right, leading to a spanning tree of weight 1. ◀

4.3 A Tighter Analysis

While showing tightness in the general case, the lower-bound proof in Theorem 6 is based on a graph with a maximum spanning tree of a very small weight. In this section we show that g_{max} approximates π^*_{max} better when $w(MST(G))$ is large.

▶ **Theorem 7.** *Let $G = \langle V, E, w\rangle$ be a weighted graph, and assume that the weights in G are normalized such that the maximum weight of an edge in E is 1. Then, g_{max} is a $1.5 + \frac{1}{w(MST(G))}$-max-approximation of π^*_{max}.*

Proof. We start with a brief description of the main idea of the proof. Let $\langle e_1, \ldots, e_{n-1}\rangle$ be the edges claimed by MAX and MIN in this order when MAX follows a greedy strategy g_{max} and MIN follows a strategy π_{min} that is optimal against g_{max}. Using the fact that g_{max} is a greedy strategy, we will show that MIN has a strategy π'_{min} such that, when pitted against an optimal strategy π^*_{max} of MAX (in fact, against any max strategy), it ensures that the weight of the resulting spanning tree is at most $(1.5+1/w(MST(G)))\cdot\sum_{i=1}^{n-1} w(e_i)$. Note that π'_{min} might not be an optimal min strategy, but this only makes the proven result stronger. The heart of the argument is that as long as MAX can claim high (in comparison to what she claimed when she followed g_{max}) weight edges, MIN can claim quite a few low (in comparison to what he claimed when he followed π_{min}) weight edges.

We proceed to the formal proof. Let $\pi_{min} \in \Pi_{min}$ be a min strategy for which $val_{max}(g_{max}) = w(g_{max}, \pi_{min})$. Let $\langle e_1, \ldots, e_{n-1}\rangle$ be a vector of edges of $T(g_{max}, \pi_{min})$, where, for every $1 \le i \le n-1$, if i is odd, then e_i is chosen by MAX in her $((i+1)/2)$-th move, and if i is even, then e_i is chosen by MIN in his $(i/2)$-th move. Let $E_{odd} = \{e_1, e_3, \ldots, e_b\}$, where $b = n-1-(n \mod 2)$, be the edges chosen by MAX, and let $E_{even} = \{e_2, e_4, \ldots, e_a\}$, where $a = n-2+(n \mod 2)$, be the edges chosen by MIN. Let $d_1 > \ldots > d_k$ be the distinct weights of the edges in E_{odd}, and let $t_1, \ldots, t_k$ be positive integers such that E_{odd} contains exactly t_i edges of weight d_i for every $1 \le i \le k$. Let $t'_0 = 0$ and, for every $1 \le i \le k$, let $t'_i = t'_{i-1} + 2t_i$. Thus, $t'_i = \sum_{j=1}^{i} 2t_j$. Note that, for every $1 \le i \le k$, the edges of E_{odd} whose weight is d_i are $\{e_{t'_{i-1}+1}, e_{t'_{i-1}+3}, \ldots, e_{t'_i-1}\}$. For example, $w(e_1) = w(e_3) = \ldots = w(e_{2t_1-1}) = d_1$, and $w(e_{2t_1+1}) = w(e_{2t_1+3}) = \ldots = w(e_{2t_1+2t_2-1}) = d_2$. Since the weights in G are normalized so that the maximum weight of an edge in G is 1 and since g_{max} is greedy, we have that $d_1 = 1$.

We argue that MIN has a strategy π'_{min} with which he can ensure that, by deviating from the greedy strategy g_{max}, MAX does not greatly improve the weight of the tree she builds with him. We define the strategy π'_{min} as follows. Consider a forest $F_m = \{e'_1, e'_2, \ldots, e'_m\} \in \mathcal{F}^{odd}_G$, where $m < \lfloor\frac{n-1}{2}\rfloor$. Let $0 \le i < k$ be the unique integer for which $\frac{t'_i}{2} \le m < \frac{t'_{i+1}}{2}$. Then, $\pi'_{min}(F_m)$ is an arbitrary edge in $M(F_m) \cap \{e_2, e_4, \ldots, e_{t'_{i+1}}\}$; by definition, this is a legal move. Moreover, by the independent set exchange property of the cycle matroid of G, such an edge exists. For example, if $m < t_1$, then $\pi'_{min}(F_m)$ is an arbitrary edge of $\{e_2, e_4, \ldots, e_{2t_1}\}$ that was not chosen earlier and does not close a cycle with F_m.

Since $val_{max}(\pi^*_{max}) \le w(\pi^*_{max}, \pi'_{min})$, it suffices to prove that $\frac{w(\pi^*_{max},\pi'_{min})}{val_{max}(g_{max})} \le 1.5 + \frac{1}{w(MST(G))}$. For an integer t, let $V^t_1, \ldots, V^t_{s_t}$ be the vertex sets of the connected components induced by the forest $\{e_1, \ldots, e_t\}$. Let E^t denote the set of edges of G that are contained in some connected component of $\{e_1, \ldots, e_t\}$, that is, $\langle u, v\rangle \in E^t$ if and only if there exists some $1 \le i \le s_t$ such that $u, v \in V^t_i$. Note that every forest in G contains at most $\sum_{j=1}^{s_t}(|V^t_j| - 1) = t$ edges of E^t.

Let $E' = \{e'_1, \ldots, e'_{n-1}\}$ denote the edge set of $T(\pi^*_{max}, \pi'_{min})$. Note that by the description of the strategy π'_{min}, for every $1 \le i < k$, the forest $\{e'_1, e'_2, \ldots, e'_{t'_i/2}\}$ contains at least $\lfloor\frac{t'_i/2}{2}\rfloor$ edges from $E^{t'_i} \cap E_{even}$. Since $E' \cap E^{t'_i}$ contains at most t'_i edges, it follows that

$E' \cap E^{t'_i}$ contains at most $t'_i - \lfloor \frac{t'_i/2}{2} \rfloor = \lceil 1.5 \cdot \frac{t'_i}{2} \rceil$ edges from $E \setminus E_{even}$. Note that for every edge $e \notin E^{t'_i}$, we have that $w(e) \leq d_{i+1}$. Indeed, otherwise MAX would have chosen $e_{t'_i+1}$ such that $w(e_{t'_i+1}) > d_{i+1}$. Hence, $E' \setminus E_{even}$ contains at most $1.5 \cdot \frac{t'_i}{2} + 0.5$ edges from $\{e \in E : w(e) > d_{i+1}\}$.

We now show that $E' \setminus E_{even}$ contains at most $1.5 \cdot \frac{t'_k}{2} + 0.5$ edges. Assume first that $n-1$ is even and thus $t'_k = n-1$. The forest $\{e'_1, e'_2, \ldots, e'_{t'_k/2}\}$ contains at least $\left\lfloor \frac{t'_k/2}{2} \right\rfloor$ edges from E_{even}. Since E' contains t'_k edges, it follows that E' contains at most $t'_k - \lfloor \frac{t'_k/2}{2} \rfloor = \lceil 1.5 \cdot \frac{t'_k}{2} \rceil$ edges from $E \setminus E_{even}$. Hence, $E' \setminus E_{even}$ contains at most $1.5 \cdot \frac{t'_k}{2} + 0.5$ edges. Now, assume that $n-1$ is odd and thus $t'_k = n$. Note that E' contains at least $\lfloor \frac{\lfloor \frac{n-1}{2} \rfloor}{2} \rfloor = \lfloor \frac{0.5n-1}{2} \rfloor$ edges from E_{even}. Therefore, the size of $E' \setminus E_{even}$ is at most $n-1-\lfloor \frac{0.5n-1}{2} \rfloor = \lceil n-1-\frac{0.5n-1}{2} \rceil = \lceil \frac{3n}{4} - 0.5 \rceil \leq \lceil \frac{3n}{4} \rceil = \lceil 1.5 \cdot \frac{t'_k}{2} \rceil \leq 1.5 \cdot \frac{t'_k}{2} + 0.5$.

Since for every $1 \leq i < k$ the forest $E' \setminus E_{even}$ contains at most $1.5 \cdot \frac{t'_i}{2} + 0.5$ edges from $\{e \in E : w(e) > d_{i+1}\}$, and since $E' \setminus E_{even}$ contains at most $1.5 \cdot \frac{t'_k}{2} + 0.5$ edges, then the total weight of $E' \setminus E_{even}$ is at most $d_1(1.5 \cdot \frac{t'_1}{2} + 0.5) + \sum_{i=2}^k d_i \cdot [(1.5 \cdot \frac{t'_i}{2} + 0.5) - (1.5 \cdot \frac{t'_{i-1}}{2} + 0.5)] = d_1(1.5t_1 + 0.5) + \sum_{i=2}^k d_i \cdot (1.5t_i) = 0.5d_1 + \sum_{i=1}^k 1.5t_i d_i$.

We are now ready to bound $\frac{w(\pi^*_{max}, \pi'_{min})}{val_{max}(g_{max})}$ from above.

$$
\begin{aligned}
& \frac{w(\pi^*_{max}, \pi'_{min})}{val_{max}(g_{max})} = \frac{w(E')}{w(E_{even}) + \sum_{i=1}^k t_i d_i} \leq \frac{w(E_{even}) + w(E' \setminus E_{even})}{w(E_{even}) + \sum_{i=1}^k t_i d_i} \\
\leq\ & \frac{w(E_{even}) + 0.5d_1 + \sum_{i=1}^k 1.5t_i d_i}{w(E_{even}) + \sum_{i=1}^k t_i d_i} = \frac{w(E_{even}) + \sum_{i=1}^k t_i d_i + \sum_{i=1}^k 0.5t_i d_i + 0.5d_1}{w(E_{even}) + \sum_{i=1}^k t_i d_i} \\
\leq\ & 1 + \frac{\sum_{i=1}^k 0.5t_i d_i + 0.5d_1}{\sum_{i=1}^k t_i d_i} = 1.5 + \frac{0.5d_1}{\sum_{i=1}^k t_i d_i} \leq 1.5 + \frac{0.5}{0.5 \cdot w(MST(G))} \\
=\ & 1.5 + \frac{1}{w(MST(G))}.
\end{aligned}
$$

The last inequality follows from the fact $\sum_{i=1}^k t_i d_i \geq 0.5 \cdot w(MST(G))$ (see proof of Theorem 4) and $d_1 = 1$. ◀

The following theorem asserts that the approximation ratio given in Theorem 7 is tight.

▶ **Theorem 8.** *Let $n \geq 1$ be an odd integer. There exists a weighted graph G_n with $w(MST(G_n)) = 2n$ and with a maximum edge weight of 1, such that $\frac{val_{max}(\pi^*_{max})}{val_{max}(g_{max})} = 1.5 + \frac{1}{w(MST(G))}$.*

Proof. We define $G_n = \langle V, E, w \rangle$ as follows. First, let $V = V_1 \cup V_2$, where $V_1 = \{v_0, v_1, \ldots, v_n\}$ and $V_2 = \{v_0, u_1, \ldots, u_n\}$. Note that the vertex v_0 appears in both V_1 and V_2. Then, let $E = E_1 \cup E_2$ where $E_1 = \{\langle v_i, v_{i+1} \rangle : 0 \leq i \leq n-1\}$ and $E_2 \subseteq V_2 \times V_2$ is the disjoint union of two spanning trees T_0 and T_1 on the vertices of V_2. It is not hard to see that such two spanning trees always exist. For $n \leq 2$, one needs parallel edges, as in G_1, which appears in Figure 3. For $n \geq 3$, the graph G_n appears in Figure 4, where the edges in T_1 are solid, and these in T_0 are dashed.

For every edge $e \in E_1 \cup T_1$ we have $w(e) = 1$ and for every edge $e \in T_0$ we have $w(e) = 0$. The edges in E_1 must be contained in every spanning tree of G_n. Therefore, if m edges from T_1 are chosen during the game for some $m \leq n$, then the outcome of the game is $m + n$.

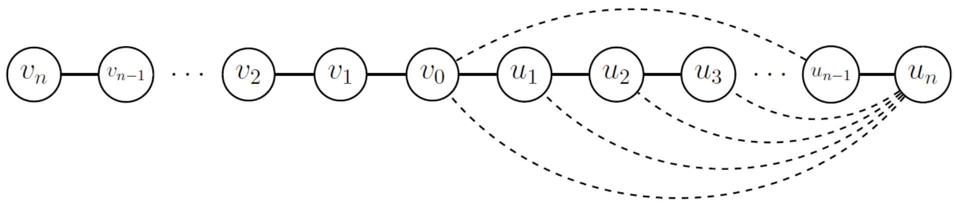

Figure 4 The graph G_n with $\frac{val_{max}(\pi^*_{max})}{val_{max}(g_{max})} = 1.5 + \frac{1}{w(MST(G_n))}$.

Thus, an optimal strategy π^*_{max} is to have as many edges from T_1 as possible. Hence, by Lemma 2 we have $val_{max}(\pi^*_{max}) = \lceil \frac{n}{2} \rceil + n$. In the strategy g_{max}, MAX chooses only the n edges in E_1, and hence $val_{max}(g_{max}) = n$.

Since n is odd, we have $\frac{val_{max}(\pi^*_{max})}{val_{max}(g_{max})} = \frac{\lceil \frac{n}{2} \rceil + n}{n} = \frac{\frac{n}{2}+0.5+n}{n} = 1.5 + \frac{1}{2n} = 1.5 + \frac{1}{w(MST(G_n))}$. ◀

5 A Stochastic Setting

The weighted graphs $\{G_n : n \in \mathbb{N}\}$ depicted in Figure 2 form an infinite family of games in which g_{max} is an optimal strategy for MAX. In this section we prove that g_{max} is not far from being optimal in a very natural and general case.

▶ **Theorem 9.** *Consider the weighted graph $G = \langle V, E, w \rangle$, where $V = [n]$, $E = \binom{[n]}{2}$, and $\{w(e) : e \in E\}$ are independent random variables, each having a uniform distribution over $[0,1]$. Then, asymptotically almost surely (a.a.s., for brevity)*

$$\lim_{n\to\infty} \frac{val_{max}(g_{max})}{val_{max}(\pi^*_{max})} = 1.$$

The main ingredient in our proof of Theorem 9 is the following result, which is an immediate corollary of the main result of [12] (see also [9] and the many references therein).

▶ **Theorem 10.** *For $n \geq 1$, consider the complete graph with n vertices K_n, and let $\{X_e : e \in E(K_n)\}$ be independent random variables, each having a uniform distribution over $[0,1]$. Let Y_m (respectively, Y_M) denote the weight of a minimum (respectively, maximum) spanning tree. Then*

(a) $\lim_{n\to\infty} Pr(Y_m \leq 1.21) = 1$.

(b) $\lim_{n\to\infty} Pr(Y_M \geq n - 2.21) = 1$.

Proof of Theorem 9. It readily follows from Theorem 4 and Part (b) of Theorem 10 that a.a.s. $val_{max}(g_{max}) \geq (n-2.21)/2$. Let T be a spanning tree with weight at most 1.21; such a tree exists a.a.s. by Part (a) of Theorem 10. It follows by Lemma 2 that MIN has a strategy to ensure that the tree he builds with MAX contains at least $\lfloor |T|/2 \rfloor = \lfloor (n-1)/2 \rfloor$ edges of T. The weight of the tree they build is thus at most $1.21 + \lceil (n-1)/2 \rceil \leq (n+2.42)/2$. Hence, a.a.s.

$$\lim_{n\to\infty} \frac{val_{max}(g_{max})}{val_{max}(\pi^*_{max})} \geq \lim_{n\to\infty} \frac{(n-2.21)/2}{(n+2.42)/2} = 1$$

as claimed. ◀

6 A Two-Turn Variant of the Spanning-Tree Game

In this section we study a variant of the game in which the players alternate turns only once. Formally, we have the following. A *game* is a pair $\langle G, k\rangle$, where $G = \langle V, E, w\rangle$ is a weighted graph with n vertices and $1 \le k \le n-1$ is an integer. In a game on $\langle G, k\rangle$, first MAX chooses a forest $F \subseteq E$ of size k. Then, MIN complements F to a spanning tree of G by choosing $n-1-k$ edges. MAX wants to maximize the weight of the resulting spanning tree and MIN aims to minimize it. Let $g_{max} \subseteq E$ be a strategy for MAX in which she chooses a forest of size k with a maximum weight, that is, MAX chooses a forest in a greedy manner. Note that while we still use the notation which was introduced in Subsection 2.4 (e.g., g_{max}), the definition of a strategy is different in this setting. A strategy π_{max} of MAX is simply the edge set of some forest of G of size k. Similarly, a strategy π_{min} for MIN is a function that, given a forest F of size k, returns a forest F' of size $n-1-k$ such that $F \cup F'$ is a spanning tree.

▶ **Theorem 11.** *Let $\langle G, k\rangle$ be a game, where $G = \langle V, E, w\rangle$ and $|V| = n$. Then, $val_{max}(g_{max}) \ge \frac{k}{n-1} \cdot w(MST(G))$.*

Proof. Let $T = \{e_1, \ldots, e_{n-1}\}$, where $w(e_1) \ge \ldots \ge w(e_{n-1})$, be an MST obtained by complementing g_{max} in a greedy manner. That is, $g_{max} = \{e_1, \ldots, e_k\}$. Note that for every $k < i \le n-1$ we have $w(e_i) \le w(e_k)$. Therefore, $w(MST(G)) = w(T) = w(\{e_1, \ldots, e_k\}) + w(\{e_{k+1}, \ldots, e_{n-1}\}) \le w(g_{max}) + (n-k-1) \cdot w(e_k)$. Since $w(e_k) \le \frac{1}{k} \cdot w(g_{max})$, we have $w(MST(G)) \le w(g_{max}) + (n-k-1) \cdot \frac{1}{k} \cdot w(g_{max}) = \frac{n-1}{k} \cdot w(g_{max}) \le \frac{n-1}{k} \cdot val_{max}(g_{max})$. ◀

▶ **Theorem 12.** *Let $\langle G, k\rangle$ be a game, where $G = \langle V, E, w\rangle$ and $|V| = n$. Then, g_{max} is a 2-max-approximation.*

Proof. Let π_{min} be a strategy for which $val_{max}(g_{max}) = w(g_{max}, \pi_{min})$ and let $T = T(g_{max}, \pi_{min})$. Let π^*_{max} be an optimal strategy for MAX. Consider the strategy π'_{min} of MIN in which π^*_{max} is complemented to a spanning tree as follows. Since $|\pi^*_{max}| = k$ and $|T| = n-1$, MIN can choose $n-1-k$ edges from T due to the independent set exchange property of the cycle matroid of G. For such a strategy π'_{min}, we have $val_{max}(\pi^*_{max}) \le w(\pi^*_{max}, \pi'_{min}) \le w(\pi^*_{max}) + w(T)$. Since g_{max} is a forest of maximum weight among all forests of G with k edges, it follows that $w(\pi^*_{max}) \le w(g_{max})$, and thus $val_{max}(\pi^*_{max}) \le w(g_{max}) + w(T) \le 2 \cdot w(T) = 2 \cdot val_{max}(g_{max})$. ◀

The following result is a straightforward consequence of Theorems 11 and 12.

▶ **Corollary 13.** *Let $\langle G, k\rangle$ be a game, where $G = \langle V, E, w\rangle$ and $|V| = n$. Then, g_{max} is a $\min\{2, \frac{n-1}{k}\}$-max-approximation.*

In the following theorem we show that the approximation ratio in Corollary 13 is tight.

▶ **Theorem 14.** *Let $n > 1$ and $1 \le k \le n-1$ be integers. There exists a game $\langle G, k\rangle$, where $G = \langle V, E, w\rangle$ and $|V| = n$, such that $\frac{val_{max}(\pi^*_{max})}{val_{max}(g_{max})} = \min\{2, \frac{n-1}{k}\}$, where π^*_{max} is an optimal strategy for MAX in G.*

Proof. Let $V = V_1 \cup V_2$, where $V_1 = \{v_0, v_1, \ldots, v_k\}$ and $V_2 = \{v_0, u_1, \ldots, u_{n-1-k}\}$. Note that the vertex v_0 appears in both V_1 and V_2. Let $E = E_1 \cup E_2$, where $E_1 = \{\langle v_i, v_{i+1}\rangle : 0 \le i \le k-1\}$ and $E_2 = E(T_0) \cup E(T_1)$, where T_0 and T_1 are edge-disjoint spanning trees of $G[V_2]$ (we allow parallel edges in E_2). For every edge $e \in E_1 \cup T_1$ we set $w(e) = 1$ and for every edge $e \in T_0$ we set $w(e) = 0$. Note that if MAX chooses m edges in T_1 for some $m \le n-1-k$, then MIN can choose $n-1-k-m$ edges in T_0 due to the independent set exchange property of the cycle matroid of G. The edges of E_1 must be contained in every spanning tree of G. Therefore, if MAX chooses m edges from T_1, then the outcome of the game is $m+k$. Thus, the optimal strategy π^*_{max} contains as many edges from T_1 as possible, namely, $\min\{k, n-1-k\}$ edges from T_1. The strategy g_{max} contains the k edges in E_1, and therefore $val_{max}(g_{max}) = k$.

If $k \leq \frac{n-1}{2}$ then π^*_{max} contains k edges from T_1 and hence we have $\frac{val_{max}(\pi^*_{max})}{val_{max}(g_{max})} = \frac{2k}{k} = 2 = \min\{2, \frac{n-1}{k}\}$. If $k > \frac{n-1}{2}$ then π^*_{max} contains $n-1-k$ edges from T_1 and hence we have $\frac{val_{max}(\pi^*_{max})}{val_{max}(g_{max})} = \frac{n-1}{k} = \min\{2, \frac{n-1}{k}\}$. ◀

7 Discussion

We studied a game variant of the classic maximum spanning-tree problem. Both the classic problem and our spanning-tree game can be generalized in a straightforward way to all matroids. In the game setting, given a weighted matroid $M = \langle E, \mathcal{I}, w\rangle$, MAX and MIN alternate turns in claiming elements of E while ensuring that the set of elements claimed so far by both players is in $\mathcal{I}$. The game is over as soon as the set of claimed elements is a basis B of M. MAX aims to maximize the total weight of B and MIN aims to minimize it. It is not hard to show that all of our results (with the exception of Theorem 9, which deals only with weighted complete graphs) apply in this more general setting. The only non-trivial generalization is that of one specific point in the proof of Theorem 7, which we explain below.

When defining E^t, instead of relying on the connected components of the forest $\{e_1, \ldots, e_t\}$, one can use the rank function[5] r of the matroid. That is, $E^t = \{e \in E : r(\{e\} \cup \{e_1, \ldots, e_t\}) = r(\{e_1, \ldots, e_t\})\}$. It then readily follows from the definitions of r and of E^t that $|B \cap E^t| \leq t$ holds for every $B \in \mathcal{I}$.

The graph depicted in Figure 3, which is used to show that, in general, the competitive ratio of greedy strategies is 2, contains parallel edges. One then wonders whether the competitive ratio of greedy strategies is better than 2 under the assumption that the graph on which the game is played is simple. At the moment we only know that this ratio is between 5/3 and 2. One can also consider graphs that are not only simple, but have a large girth[6]. The intuition behind this is that, in order to prevent MAX from claiming a certain edge, MIN must ensure that claiming it closes a cycle, and this seems harder if all cycles are long. Moreover, when the girth is 2, i.e., there are parallel edges, we know that the competitive ratio is 2. On the other hand, when the game is played on a tree, i.e., the girth is infinite, the competitive ratio is trivially 1. This shows that increasing the girth does decrease (in some way) the competitive ratio of greedy strategies from 2 to 1.

Finally, our game is a special case of the so-called *biased game*, in which MAX claims p edges per turn and then MIN claims q edges per turn, where p and q are positive[7] integers that are allowed to grow with n. It would be interesting to study how changing the parameters p and q would affect our results.

[5] The rank function of a matroid $M = \langle E, \mathcal{I}\rangle$ is a mapping $r : 2^E \to \mathbb{N}$ that maps each subset A of E to the size of a largest independent set it contains; i.e., $r(A) = \max\{|B| : B \subseteq A, B \in \mathcal{I}\}$.

[6] The girth of a graph G is the length of a shortest cycle in G. If G is a forest, then its girth is defined to be ∞.

[7] In fact, by allowing $p = 0$ (respectively, $q = 0$) we get the original minimum (resp., maximum) spanning tree problem for which greedy strategies are optimal regardless of the value of q (resp., p).

References

1. T. Bartnicki, J. A. Grytczuk, H. A. Kierstead, and X. Zhu. The map coloring game. *American Mathematical Monthly*, 114:793–803, 2007.
2. C.G. Bird. On cost allocation for a spanning tree: a game theoretic approach. *Networks*, 6(4):335–350, 1976.
3. J. Bruno and L. Weinberg. A constructive graph-theoretic solution of the shannon switching game. *IEEE Transactions on Circuit Theory*, 17(1):74–81, 1970.
4. J. Cardinal, E.D. Demaine, S. Fiorini, G. Joret, S. Langerman, I. Newman, and O. Weimann. The stackelberg minimum spanning tree game. *Algorithmica*, 59(2):129–144, 2011.
5. J. Cardinal, E.D. Demaine, S. Fiorini, G. Joret, I. Newman, and O. Weimann. The stackelberg minimum spanning tree game on planar and bounded-treewidth graphs. *Journal of combinatorial optimization*, 25(1):19–46, 2013.
6. A.K. Chandra, D.C. Kozen, and L.J. Stockmeyer. Alternation. *Journal of the Association for Computing Machinery*, 28(1):114–133, 1981.
7. A. Claus and D. J. Kleitman. Cost allocation in networks: The bulk supplier problem. *Networks*, 4(1):1–17, 1974.
8. S.A. Cook. Path systems and language recognition. In *Proc. 2nd ACM Symp. on Theory of Computing*, pages 70–72, 1970.
9. C. Cooper, A. Frieze, N. Ince, S. Janson, and J. Spencer. On the length of a random minimum spanning tree. *Combinatorics, Probability and Computing*, 25:89–107, 2016.
10. T.H. Cormen, C.E. Leiserson, and R.L. Rivest. *Introduction to Algorithms*. MIT Press and McGraw-Hill, 1990.
11. B. Dutta and A. Kar. Cost monotonicity, consistency and minimum cost spanning tree games. *Games and Economic Behavior*, 48(2):223–248, 2004.
12. A. M. Frieze. On the value of a random minimum spanning tree problem. *Discrete Applied Mathematics*, 10:47–56, 1985.
13. Z. Füredi, D. Reimer, and A. Seress. Triangle-free game and extremal graph problems. *Congressus Numerantium*, 82:123–128, 1991.
14. R.L Graham and P. Hell. On the history of the minimum spanning tree problem. *Annals of the History of Computing*, 7(1):43–57, 1985.
15. D. Granot and G. Huberman. Minimum cost spanning tree games. *Mathematical programming*, 21(1):1–18, 1981.
16. D. Granot and G. Huberman. On the core and nucleolus of minimum cost spanning tree games. *Mathematical programming*, 29(3):323–347, 1984.
17. D. Hefetz, M. Krivelevich, A. Naor, and M. Stojaković. On saturation games. *European Journal of Combinatorics*, 41:315–335, 2016.
18. A. Kar. Axiomatization of the shapley value on minimum cost spanning tree games. *Games and Economic Behavior*, 38(2):265–277, 2002.
19. J.B. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical society*, 7(1):48–50, 1956.
20. O. Kupferman. Examining classical graph-theory problems from the viewpoint of formal-verification methods. In *Proc. 49th ACM Symp. on Theory of Computing*, page 6, 2017.
21. O. Kupferman, G. Vardi, and M.Y. Vardi. Flow games. In *Proc. 37th Conf. on Foundations of Software Technology and Theoretical Computer Science*, 2017, to appear.
22. A. Lehman. A solution of the shannon switching game. *Journal of the Society for Industrial and Applied Mathematics*, 12(4):687–725, 1964.
23. O. Lichtenstein and A. Pnueli. Checking that finite state concurrent programs satisfy their linear specification. In *Proc. 12th ACM Symp. on Principles of Programming Languages*, pages 97–107, 1985.

24 N. Nisan, T. Roughgarden, E. Tardos, and V.V. Vazirani. *Algorithmic Game Theory*. Cambridge University Press, 2007.

25 J. Oxley. *Matroid Theory, 2nd edition*. Oxford University Press, 2011.

26 R.C. Prim. Shortest connection networks and some generalizations. *Bell Labs Technical Journal*, 36(6):1389–1401, 1957.

27 L.J. Stockmeyer. On the combinational complexity of certain symmetric boolean functions. *Mathematical Systems Theory*, 10:323–336, 1977.

28 R.E. Tarjan. *Data structures and network algorithms*. SIAM, 1983.