FULL LENGTH PAPER

Series B

An efficient characterization of submodular spanning tree games

Zhuan Khye Koh[1] · Laura Sanità[2,3]

Abstract

Cooperative games form an important class of problems in game theory, where a key goal is to distribute a value among a set of players who are allowed to cooperate by forming coalitions. An outcome of the game is given by an allocation vector that assigns a value share to each player. A crucial aspect of such games is *submodularity* (or *convexity*). Indeed, convex instances of cooperative games exhibit several nice properties, e.g. regarding the existence and computation of allocations realizing some of the most important solution concepts proposed in the literature. For this reason, a relevant question is whether one can give a polynomial-time characterization of submodular instances, for prominent cooperative games that are in general non-convex. In this paper, we focus on a fundamental and widely studied cooperative game, namely *the spanning tree game*. An efficient recognition of submodular instances of this game was not known so far, and explicitly mentioned as an open question in the literature. We here settle this open problem by giving a polynomial-time characterization of submodular spanning tree games.

✉ Zhuan Khye Koh
  z.koh3@lse.ac.uk

  Laura Sanità
  lsanita@uwaterloo.ca

[1] Department of Mathematics, London School of Economics and Political Science, London, UK

[2] Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Canada

[3] Department of Mathematics and Computer Science, Eindhoven University of Technology, Eindhoven, Netherlands

Mathematics Subject Classification 05C05 Trees · 05C57 Games on graphs (graph-theoretic aspects) · 91A12 Cooperative games

1 Introduction

Cooperative games are among the most studied classes of problems in game theory, with plenty of applications in economics, mathematics, and computer science. In such games, a central question is how to distribute cost (or revenue) among a set of participants, usually called *players*, who are allowed to cooperate. Formally, we are given a set N of players, and a characteristic function $v : 2^N \to \mathbb{R}$, with $v(\emptyset) = 0$. Here, $v(S)$ represents the cost paid (revenue received) by the subset S of players if they choose to form a coalition. An outcome of the game is given by an *allocation* $y \in \mathbb{R}^N$ such that $\sum_{v \in N} y_v = v(N)$, which assigns a cost (revenue) share to each player. Of course, there are a number of criteria for evaluating how "good" an allocation is, such as *stability*, *fairness*, and so on.

Probably the most popular solution concept for cooperative games is the *core*. It is the set of stable outcomes where no subset of players has an incentive to form a coalition to deviate. In a cooperative cost game, this translates naturally to the following constraint: $\sum_{v \in S} y_v \leq v(S)$, for all $S \subseteq N$. Intuitively, if this constraint is violated for some set S, the total cost currently paid by the players in S is more than the total cost $v(S)$ they would have to pay if they form a coalition—this incentivizes these players to deviate from the current allocation. Besides the core, there are several other crucial solution concepts which have been defined in the literature, e.g. the *Shapley value*, the *nucleolus*, the *kernel*, the *bargaining set*, and the *von Neumann-Morgenstern solution sets* (we refer to [2] for details). Many fundamental questions involving such solution concepts have been investigated in the past few decades: Which cooperative game instances admit an allocation realizing a particular solution concept? Can we efficiently compute it? Can we test whether a given allocation belongs to such sets?

Submodularity (or *convexity*) is a crucial property which yields interesting answers to some of the questions above. An instance of a cooperative cost game is called submodular if the characteristic function v is submodular, meaning that

$$\forall A, B \subseteq N, \ v(A) + v(B) \geq v(A \cup B) + v(A \cap B). \qquad (*)$$

Submodular games exhibit a large number of desirable properties. In particular, (1) the core is always non-empty and an allocation in the core can be computed in polynomial time [13]; (2) testing whether an allocation belongs to the core is equivalent to separating over the extended polymatroid of v, which can be performed efficiently [8]; (3) computing the nucleolus can be done efficiently [10]; (4) there is a nice "snowballing" effect that arises when the game is played cooperatively, meaning that joining a coalition becomes more attractive as the coalition grows, and so the value of the so-called grand coalition $v(N)$ is always reached [13]. We refer to [11,13] for other interesting properties of submodular games involving other crucial solution concepts. Given these observations, it is not surprising that some researchers have investigated whether it is possible to give an efficient characterization of submodular instances, for

prominent cooperative games that are in general non-convex. Such characterizations are known, for example, for the minimum coloring game and the minimum vertex cover game [12], as well as for some communication games [15].

This paper focuses on one of the most fundamental cooperative games, namely the *spanning tree game*. This game was introduced more than 40 years ago [1,3], and since then it has been widely studied in the literature. To get an intuition about the problem, consider the following setting. A set N of clients would like to be connected to a central source r which can provide a service to them. The clients wish to build a network connecting them to the source r, at minimum cost. An obvious way to solve this problem is to compute a minimum spanning tree connecting $N \cup \{r\}$. But how should the clients fairly split the cost of the tree among them? Formally, an instance of the spanning tree game is described by an edge-weighted complete graph $G = (V, E)$ where $V = N \cup \{r\}$. The set of players is given by N, and the value $v(S)$ of the characteristic function is equal to the cost of a minimum spanning tree in the subgraph induced by $S \cup \{r\}$.

Despite being one of the most studied cooperative games, the existence of an efficient characterization of submodularity for the spanning tree game has remained elusive so far. Granot and Huberman [7] proved that spanning tree games are *permutationally convex* (which is a generalization of submodularity). Their result implies that the core is always non-empty for such games, despite being non-convex in general (this was first proven by the same authors in [6]). However, other nice properties of submodular games do not generalize: for general spanning tree games, testing core membership is coNP-hard [4], and computing the nucleolus is NP-hard [5]. Trudeau [14] gave a sufficient condition for an instance of the game to be submodular. An important step forward was made by Kobayashi and Okamoto [9], who gave a characterization of submodularity for instances of the spanning tree game where the edge weights are restricted to take only two values. For general weights, they stated some necessary (but not always sufficient) as well as some sufficient (but not always necessary) conditions for an instance to be submodular. Whether a polynomial-time characterization of submodularity exists for spanning tree games is left as an open question. In fact, they stated twice in their paper:
"We feel that recognizing a submodular minimum-cost spanning tree game is coNP-complete, but we are still far from proving such a result."

Our results and techniques. In this paper, we finally settle this open question: we give a polynomial-time characterization of submodular spanning tree games.

Our characterization uses combinatorial techniques and it is based on two main ingredients. The first one, described in Sect. 3, is a generalization of Kobayashi and Okamoto's result [9]. When the edges can have only two distinct weights, they proved that the only obstruction to submodularity comes from the presence of certain cycles in the graph induced by the cheaper edges. When dealing with more weight values, say $w_1 < w_2 < \cdots < w_k$, things become necessarily more complicated. We can still prove that an obstruction to submodularity is given by certain cycles, which we call *violated*, but (a) our definition of violated cycles is more involved than the one in [9], and (b) we have to look for such cycles not just in one induced graph, but in each graph induced by the edges of weight at most w_i, for all $i < k$.

Furthermore, the presence of violated cycles is not anymore the only obstruction to submodularity. Roughly speaking, violated cycles capture how the edges of a certain weight should relate to the cheaper ones, but we still need a condition that takes into account the "magnitude" of distinct weight values, when $k > 2$. This leads to the second main ingredient of our characterization, described in Sect. 4. We show that, under the assumption of not having violated cycles, we can identify polynomially many subsets of vertices which could yield the highest possible violation to the submodularity inequality (*). We can then efficiently test the submodularity of our instance by checking whether the inequality (*) is satisfied on this family of subsets of vertices. Combining these two ingredients yields a polynomial-time characterization of submodularity for spanning tree games, as described in Sect. 5.

We conclude our paper with an additional result. As previously mentioned, the authors of [9] gave a necessary condition for submodularity of the spanning tree game. They also stated that they do not know whether their condition can be verified in polynomial time. We answer this question affirmatively in Sect. 6.

2 Preliminaries and notation

For a subset $S \subseteq V$, denote $\mathsf{mst}(S)$ as the weight of a minimum spanning tree in $G[S]$, where $G[S]$ is the subgraph of G induced by S. For a subset $F \subseteq E$, denote $w(F)$ as the sum of edge weights in F, i.e. $w(F) := \sum_{e \in F} w(e)$. Let H be a subgraph of G. For a vertex $u \in V$ and an edge $e \in E$, the subgraphs $H \backslash u$ and $H \backslash e$ are defined as $H \backslash u := H[V(H) \backslash u]$ and $H \backslash e := (V(H), E(H) \backslash e)$. Given a vertex $u \in V$, $N_H(u)$ is the neighborhood of u in H, while $\delta_H(u)$ is the set of edges incident to u in H. Note that $u \notin N_H(u)$. For any positive integer k, $[k]$ represents the set $\{1, 2, \ldots, k\}$. Given a pair of vertices $u, v \in N$, let $\mathcal{S}_{uv}$ denote the family of vertex subsets which contain r but not u or v, i.e. $\mathcal{S}_{uv} := \{S \subseteq V : r \in S \text{ and } u, v \notin S\}$. Define the function $f_{uv} : \mathcal{S}_{uv} \to \mathbb{R}$ as

$$f_{uv}(S) := \mathsf{mst}(S \cup u) + \mathsf{mst}(S \cup v) - \mathsf{mst}(S) - \mathsf{mst}(S \cup \{u, v\}).$$

It is easy to see that the spanning tree game on G is submodular if and only if $f_{uv}(S) \geq 0$ for all $u, v \in N$ and $S \in \mathcal{S}_{uv}$. Let $w_1 < w_2 < \cdots < w_k$ be the edge weights of G. For each $i \in [k]$, define the graph $G_i := (V, E_i)$ where $E_i := \{e \in E : w(e) \leq w_i\}$. Note that $G_k = G$. For a vertex $u \in V$, denote $N_i(u)$ as the neighborhood of u in G_i. For an edge $uv \in E$, define the neighborhood of uv in G_i as

$$N_i(uv) := N_i(u) \cap N_i(v).$$

It represents the set of vertices whose edges to u and v have weight at most w_i. Notice that $u, v \notin N_i(uv)$. We will also need the following graph theory terminology. A *hole* is an induced cycle with at least four vertices. A *diamond* is the complete graph K_4 minus one edge. We will refer to the vertices of degree 2 in a diamond as *tips*. Lastly, the following property of minimum spanning trees will be useful to us.
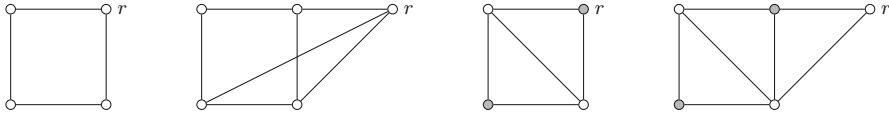
Fig. 1 The first two graphs are examples of bad holes, whereas the last two graphs are examples of bad induced diamonds. The tips of the diamonds are shaded. Every edge here has the same weight

Lemma 1 *Let T be a minimum spanning tree of G. For every subset $S \subseteq V$, there exists a minimum spanning tree of $G[S]$ which contains $E(T[S])$.*

Proof Let T_S be a minimum spanning tree of $G[S]$. We proceed by induction on $|E(T[S])\backslash E(T_S)|$. For the base case, if $|E(T[S])\backslash E(T_S)| = 0$, then T_S contains all the edges in $E(T[S])$. For the inductive step, assume $|E(T[S])\backslash E(T_S)| > 0$. Then, there exists an edge $e \in E(T[S])$ where $e \notin E(T_S)$. Adding e to T_S creates a cycle C in $G[S]$. So there exists an edge $f \in E(C)$ which is not an edge of T. Pick an appropriate f such that when added to T creates a cycle containing e. Since $T + f - e$ is a spanning tree of G, we have $w(f) \geq w(e)$. On the other hand, since $T_S + e - f$ is a spanning tree of $G[S]$, we also have $w(e) \geq w(f)$. This implies $w(e) = w(f)$, so $T_S + e - f$ is a minimum spanning tree of $G[S]$. As $|E(T[S])\backslash E(T_S + e - f)| = |E(T[S])\backslash E(T_S)| - 1$, by the induction hypothesis we are done.   □

3 Violated cycles

In this section, we will prove that a submodular spanning tree game does not contain violated cycles, which will be defined later. First, we need to introduce the concept of *well-covered* cycles.

Definition 1 Given a cycle C and a chord $f = uv$, let P_1 and P_2 denote the two u-v paths in C. The cycles $P_1 + f$ and $P_2 + f$ are called the *subcycles of C formed by f*. We say that f *covers* C if $w(f) \geq w(e)$ for all $e \in E(P_1)$ or for all $e \in E(P_2)$. If C is covered by all of its chords, then it is *well-covered*.

Next, we define the following two simple structures (see Fig. 1 for some examples). We then proceed to show that a submodular spanning tree game does not contain either of them.

Definition 2 A hole is *bad* if at least one of its vertices is not adjacent to r. An induced diamond is *bad* if its Hamiltonian cycle is well-covered but at least one of its tips is not adjacent to r.

Lemma 2 *If the spanning tree game on G is submodular, then there are no (a) bad holes or (b) bad induced diamonds in G_i for any $i < k$.*

The lemma will be proved via contrapositive separately for (a) and (b). Before jumping into the details, it is instructive to have an overview of the arguments, as they revolve around the same idea. The proof starts by assuming the existence of a bad
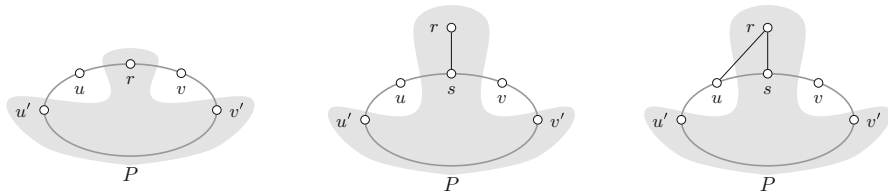
Fig. 2 An example of the subgraph $G_i[S \cup \{u, v\}]$ in Case 1 and Subcases 2.1–2.2 respectively. The shaded region represents the set S

hole or a bad induced diamond in G_i for some $i < k$. Then, from this bad structure, a subset of vertices $S \ni r$ and two additional vertices u, v will be identified such that they violate the submodularity inequality, i.e. $f_{uv}(S) < 0$.

Proof of Lemma 2(a) Let C be a bad hole in G_i for some $i < k$. Consider the following two cases.

Case 1: C contains r. Let u, v be the vertices adjacent to r in C. Define the set $S := V(C) \setminus \{u, v\}$. To prove that the instance is not submodular, it suffices to show that $f_{uv}(S) < 0$. Let P be the path obtained by deleting r, u, v from C. Let u', v' be the endpoints of P where $uu', vv' \in E(C)$ (see Fig. 2). Note that $u' = v'$ if P is a singleton. It is easy to see that

$$\mathsf{mst}(S) \geq w(P) + w_{i+1},$$
$$\mathsf{mst}(S \cup u) = w(P) + w(ru) + w(uu'),$$
$$\mathsf{mst}(S \cup v) = w(P) + w(rv) + w(vv').$$

Next, deleting the most expensive edge from C creates a minimum spanning tree of $G[S \cup \{u, v\}]$. Since this edge has weight at most w_i, we obtain

$$\mathsf{mst}(S \cup \{u, v\}) \geq w(P) + w(ru) + w(uu') + w(rv) + w(vv') - w_i.$$

Combining the equations and inequalities above yields

$$f_{uv}(S) = \mathsf{mst}(S \cup u) + \mathsf{mst}(S \cup v) - \mathsf{mst}(S) - \mathsf{mst}(S \cup \{u, v\}) \leq w_i - w_{i+1} < 0.$$

Case 2: C does not contain r. We first show that if r is adjacent in G_i to two non-adjacent vertices of C, then we are done. Since C is a bad hole in G_i, there exists a vertex $t \in V(C)$ such that $rt \notin E_i$. Starting from t, traverse the hole C in both directions until we encounter the first vertices p and q such that $rp, rq \in E_i$ respectively. Since r is adjacent to two non-adjacent vertices of C, we have $p \neq q$ and $pq \notin E(C)$. Let Q be the p-q path in C which contains t. Then, by our choice of p and q, the cycle $Q + rp + rq$ is a bad hole in G_i which contains r. It follows that the instance is not submodular by Case 1.

Thus, we may assume that r can only be adjacent to adjacent vertices of C. Pick a vertex $s \in V(C)$ with the cheapest edge to r, i.e. $w(rs) \leq w(rx)$ for all $x \in V(C)$.

Let u, v be the vertices adjacent to s in C. Define the set $S := (V(C) \cup r) \setminus \{u, v\}$. As with the previous case, it suffices to show that $f_{uv}(S) < 0$. Let P be the path obtained by deleting s, u, v from C. Let u', v' be the endpoints of P where $uu', vv' \in E(C)$ (see Fig. 2). We are left with the following two subcases.

Subcase 2.1: r is adjacent to at most one vertex of C. Since s was chosen to be the vertex in C with the cheapest edge to r, we have

$$\mathsf{mst}(S) \geq w(P) + w(rs) + w_{i+1},$$
$$\mathsf{mst}(S \cup u) = w(P) + w(rs) + w(su) + w(uu'),$$
$$\mathsf{mst}(S \cup v) = w(P) + w(rs) + w(sv) + w(vv').$$

Next, deleting the most expensive edge from C and adding rs creates a minimum spanning tree of $G[S \cup \{u, v\}]$. Since the deleted edge has weight at most w_i, we obtain

$$\mathsf{mst}(S \cup \{u, v\}) \geq w(P) + w(rs) + w(su) + w(uu') + w(sv) + w(vv') - w_i.$$

Combining the equations and inequalities above yields

$$f_{uv}(S) = \mathsf{mst}(S \cup u) + \mathsf{mst}(S \cup v) - \mathsf{mst}(S) - \mathsf{mst}(S \cup \{u, v\}) \leq w_i - w_{i+1} < 0.$$

Subcase 2.2: r is adjacent to two vertices of C. Since we assumed that r can only be adjacent to adjacent vertices of C, it follows that r is adjacent to s and either u or v in G_i. Without loss of generality, suppose that $rs, ru \in E_i$. Since s was chosen to be the vertex in C with the cheapest edge to r, we have

$$\mathsf{mst}(S) \geq w(P) + w(rs) + w_{i+1},$$
$$\mathsf{mst}(S \cup u) = w(P) + w(rs) + \min\{w(ru), w(su)\} + w(uu'),$$
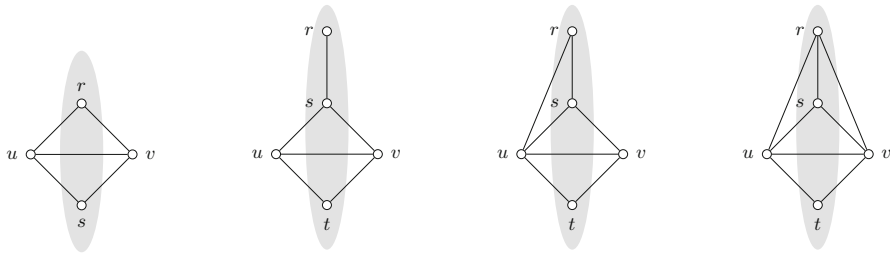$$\mathsf{mst}(S \cup v) = w(P) + w(rs) + w(sv) + w(vv').$$

Next, deleting the most expensive edge from the triangle $\{rs, ru, su\}$ does not increase the weight of a minimum spanning tree in $G[S \cup \{u, v\}]$. In fact, a minimum spanning tree can be obtained by deleting one more edge from $G_i[S \cup \{u, v\}]$. So,

$$\mathsf{mst}(S \cup \{u, v\}) \geq w(P) + w(rs) + \min\{w(ru), w(su)\} + w(uu')$$
$$+ w(sv) + w(vv') - w_i.$$

Combining the equations and inequalities above yields

$$f_{uv}(S) = \mathsf{mst}(S \cup u) + \mathsf{mst}(S \cup v) - \mathsf{mst}(S) - \mathsf{mst}(S \cup \{u, v\}) \leq w_i - w_{i+1} < 0.$$

$\square$

Fig. 3 An example of the subgraph $G_i[S \cup \{u, v\}]$ in Case 1 and Subcases 2.1–2.3 respectively. The shaded region represents the set S. For Subcase 2.1, the picture assumes that $w(rs) < w(ru)$

Proof of Lemma 2(b) Let D be a bad induced diamond in G_i for some $i < k$. Consider the following two cases.

Case 1: D contains r. Observe that r is a tip of D. Let s be the other tip and u, v be the non-tip vertices of D. Define the set $S := \{r, s\}$ (see Fig. 3). To prove that the instance is not submodular, it suffices to show that $f_{uv}(S) < 0$. It is easy to see that

$$\mathsf{mst}(S) \geq w_{i+1},$$
$$\mathsf{mst}(S \cup u) = w(ru) + w(su),$$
$$\mathsf{mst}(S \cup v) = w(rv) + w(sv).$$

Since the Hamiltonian cycle of D is well-covered, its chord uv can be deleted without increasing the weight of a minimum spanning tree in $G[S \cup \{u, v\}]$. We are now left with the Hamiltonian cycle of D, so a minimum spanning tree can be obtained by removing the most expensive edge. This gives

$$\mathsf{mst}(S \cup \{u, v\}) \geq w(ru) + w(su) + w(rv) + w(sv) - w_i.$$

Then, combining the equations and inequalities above yields

$$f_{uv}(S) = \mathsf{mst}(S \cup u) + \mathsf{mst}(S \cup v) - \mathsf{mst}(S) - \mathsf{mst}(S \cup \{u, v\}) \leq w_i - w_{i+1} < 0.$$

Case 2: D does not contain r. Let s and t be the tips of D where $w(rs) \leq w(rt)$. Note that $rt \notin E_i$ because D is a bad induced diamond in G_i. Let u and v be the non-tip vertices of D where $w(ru) \leq w(rv)$. Define the set $S := \{r, s, t\}$ (see Fig. 3). As with the previous case, it suffices to show that $f_{uv}(S) < 0$. Consider the following three subcases.

Subcase 2.1: r is adjacent to at most one vertex of D. Note that $rv \notin E_i$. It is also easy to see that

$$\mathsf{mst}(S) \geq w(rs) + w_{i+1},$$
$$\mathsf{mst}(S \cup u) = \min\{w(rs), w(ru)\} + w(su) + w(tu),$$
$$\mathsf{mst}(S \cup v) = \min\{w(rs), w(rv)\} + w(sv) + w(tv).$$

Next, observe that we can delete uv and the most expensive edge in the Hamiltonian cycle of D without increasing the weight of a minimum spanning tree in $G[S \cup \{u, v\}]$. Therefore,

$$\mathsf{mst}(S \cup \{u, v\}) \geq \min \{w(rs), w(ru)\} + w(su) + w(tu) + w(sv) + w(tv) - w_i.$$

Then, combining the equations and inequalities above yields

$$f_{uv}(S) = \mathsf{mst}(S \cup u) + \mathsf{mst}(S \cup v) - \mathsf{mst}(S) - \mathsf{mst}(S \cup \{u, v\}) \leq w_i - w_{i+1} < 0.$$

Subcase 2.2: r is adjacent to two vertices of D. We claim that if $ru, rv \in E_i$, then we are done. Note that this implies $rs, rt \notin E_i$. So, if $w(uv) \geq \max \{w(su), w(sv)\}$, then $G[\{r, s, u, v\}]$ is a bad induced diamond in G_i. Otherwise, $w(uv) \geq \max \{w(tu), w(tv)\}$ because $D \backslash uv$ is a well-covered cycle in G_i. Hence, $G[\{r, t, u, v\}]$ is a bad induced diamond in G_i. Since both diamonds contain r, the instance is not submodular by Case 1. Thus, we may assume that $rs, ru \in E_i$. Additionally, we may assume that $w(su) < \max \{w(rs), w(ru)\}$. Otherwise, $G[\{r, s, u, v\}]$ is a bad induced diamond in G_i, and we are done again by Case 1. With these two assumptions, it is then easy to see that

$$\mathsf{mst}(S) \geq w(rs) + w_{i+1},$$
$$\mathsf{mst}(S \cup u) = \min \{w(rs), w(ru)\} + w(su) + w(tu),$$
$$\mathsf{mst}(S \cup v) = w(rs) + w(sv) + w(tv).$$

Since $\mathrm{argmax} \{w(rs), w(ru)\}$ is the most expensive edge in the cycle $\{rs, ru, su\}$ by our assumption, it can be deleted without increasing the weight of a minimum spanning tree in $G[S \cup \{u, v\}]$. By a similar reasoning, uv and the most expensive edge in the Hamiltonian cycle of D can also be deleted. Therefore,

$$\mathsf{mst}(S \cup \{u, v\}) \geq \min \{w(rs), w(ru)\} + w(su) + w(tu) + w(sv) + w(tv) - w_i.$$

Finally, combining the equations and inequalities above yields

$$f_{uv}(S) = \mathsf{mst}(S \cup u) + \mathsf{mst}(S \cup v) - \mathsf{mst}(S) - \mathsf{mst}(S \cup \{u, v\}) \leq w_i - w_{i+1} < 0.$$

Subcase 2.3: r is adjacent to three vertices of D. Let $w(rv) = w_j$ for some $j \leq i$, and consider the induced diamond $G[\{r, t, u, v\}]$. If its Hamiltonian cycle is well-covered, then we are done by Case 1 because $rt \notin E_i$. So we may assume that $\max \{w(su), w(sv)\} \leq w(uv) < w(rv)$. Additionally, we may assume that $w(su) < \max \{w(rs), w(ru)\}$. Otherwise, from the previous assumption, the five edges $\{rs, ru, su, sv, uv\}$ have strictly smaller weights than $w(rv) = w_j$. Thus, $G[\{r, s, u, v\}]$ is a bad induced diamond in G_{j-1}, and we are done again by Case 1. With these two assumptions, it is then easy to see that

$$\text{mst}(S) \geq w(rs) + w_{i+1},$$
$$\text{mst}(S \cup u) = \min\{w(rs), w(ru)\} + w(su) + w(tu),$$
$$\text{mst}(S \cup v) = \min\{w(rs), w(rv)\} + w(sv) + w(tv).$$

Since rv is the most expensive edge in the cycle $\{ru, rv, su, sv\}$ by our assumption, it can be deleted without increasing the weight of a minimum spanning tree in $G[S \cup \{u, v\}]$. By a similar reasoning, argmax $\{rs, ru\}$, uv and the most expensive edge in the Hamiltonian cycle of D can also be deleted. Hence,

$$\text{mst}(S \cup \{u, v\}) \geq \min\{w(rs), w(ru)\} + w(su) + w(tu) + w(sv) + w(tv) - w_i.$$

Finally, combining the equations and inequalities above yields

$$f_{uv}(S) = \text{mst}(S \cup u) + \text{mst}(S \cup v) - \text{mst}(S) - \text{mst}(S \cup \{u, v\}) \leq w_i - w_{i+1} < 0.$$

$\square$

We are now ready to define the main object of study in this section.

Definition 3 A *violated* cycle is a well-covered cycle which contains at least one pair of non-adjacent vertices $u, v \in V$ and at least one vertex $w \in N$ not adjacent to r.

In the definition above, w can be equal to u or v. For example, if $r \in V(C)$, then the pair $\{r, w\}$ already certifies that C is a violated cycle because it can be used to satisfy both conditions. Observe that bad holes and Hamiltonian cycles of bad induced diamonds are examples of violated cycles (we consider a hole to be well-covered). The next lemma extends the scope of Lemma 2 to include violated cycles. When $k = 2$, this coincides with the condition given by Kobayashi and Okamoto [9] because every cycle in G_1 is well-covered.

Lemma 3 *If the spanning tree game on G is submodular, then there are no violated cycles in G_i for any $i < k$.*

Proof We will prove the contrapositive. Let j be the smallest integer such that G_j contains a violated cycle. By our choice of j, there are no violated cycles in G_i for all $i < j$. Let C be a smallest violated cycle in G_j. Then, $\max_{e \in E(C)} w(e) = w_j$. We first prove the following claim.

Claim 1 *For any chord f, the subcycles of C formed by f are well-covered.*

Proof Let C_1 and C_2 denote the subcycles of C formed by f. For the purpose of contradiction, suppose that C_2 is not well-covered. Let $g = uv$ be the cheapest chord in C_2 such that $w(g) < w(f)$ and $w(g) < w(h)$ for some edge $h \in E(C_2)$, where f and h lie in different subcycles of C_2 formed by g (see Fig. 4 for an example). This chord exists because C is well-covered but C_2 is not. Consider the subcycles C_3 and C_4 of C formed by g, where f is a chord of the former. Observe that any chord of C_3 covers C_3 because C is well-covered and $w(g) < w(h)$. So, the subcycle C_3 is
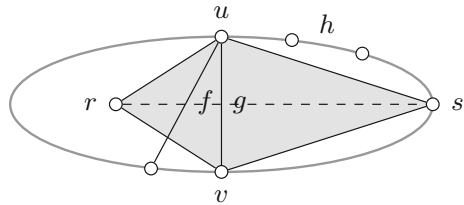
well-covered. In addition, the subcycle C_4 is well-covered because g was chosen to be the cheapest chord preventing C_2 from being well-covered.

Let $w(g) = w_\ell$ for some $\ell < j$. We claim that the subcycle C_3 is present in G_ℓ. To see this, recall that C is well-covered and $w(g) < w(h)$. Hence, it follows that $w(g) \geq w(e)$ for all $e \in E(C_3)$, which proves the claim. On the other hand, notice that the chord f is absent from G_ℓ because $w(f) > w(g)$. Since the subcycle C_3 is well-covered, its vertices are adjacent to r in G_ℓ because there are no violated cycles in G_ℓ. Next, recall that C is a violated cycle in G_j, so there exists a vertex $s \in V(C_4) \backslash V(C_3)$ such that $rs \notin E_j$. Since the subcycle C_4 is well-covered, its vertices are pairwise adjacent in G_j, as otherwise it is a smaller violated cycle than C. Note that this also implies $r \notin V(C_4)$. Thus, we have $ru, rv \in E_\ell$ and $su, sv \in E_j$.

Now, consider the 4-cycle D defined by $E(D) := \{ru, rv, su, sv\}$. It is well-covered because $w(g) = w_\ell$ and $ru, rv \in E_\ell$. Since $rs \notin E_j$, it is also a violated cycle in G_j. However, it is smaller than C. Indeed, C has at least 5 vertices because f is a chord in C_3 while g is a chord in C_2. We have arrived at a contradiction.   □

Our goal is to show the existence of a bad hole or a bad induced diamond in G_j. Then, we can invoke Lemma 2 to conclude that the game is not submodular. We may assume that C has a chord, otherwise it is trivially a bad hole. First, consider the case when $r \in V(C)$. Let $s \in V(C)$ where $rs \notin E_j$. For any chord f in C, observe that r and s lie in different subcycles of C formed by f. This is because the subcycles are well-covered by the previous claim, so the one which contains both r and s will contradict the minimality of C. Now, let g be a chord of C. Let C_r and C_s denote the subcycles of C formed by g where $r \in V(C_r)$ and $s \in V(C_s)$. Observe that the vertices of C_r are adjacent to r due to the minimality of C. Thus, C_r is a triangle. Otherwise, there is a chord in C_r incident to r, and it forms a subcycle of C which contains both r and s. On the other hand, the vertices of C_s are pairwise adjacent due to the minimality of C. Hence, C_s is also a triangle. Otherwise, there exists a chord in C_s incident to s, and it forms a subcycle of C which contains both r and s. Therefore, C is a bad induced diamond in G_j.

Next, consider the case when $r \notin V(C)$. From this point forward, we may assume that every smallest violated cycle in G_j does not contain r. Otherwise, we are back in the first case again. With this additional assumption, non-adjacency within C implies non-adjacency with r, as shown by the following claim.

Claim 2 *For any pair of vertices $u, v \in V(C)$ such that $uv \notin E_j$, we have $ru \notin E_j$ or $rv \notin E_j$.*

Fig. 5 The ellipse represents the violated cycle C in Claim 2. The shaded region highlights the violated cycle D. The dashed edges indicate non-adjacency in G_j. In this example, $u = u'$ and $v = v'$
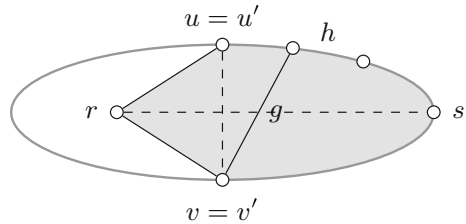
Proof For the purpose of contradiction, suppose that $ru, rv \in E_j$. Let $s \in V(C)$ such that $rs \notin E_j$. Let P_{su} and P_{sv} denote the edge-disjoint s-u and s-v paths in C respectively. Let u' and v' be the closest vertex to s on P_{su} and P_{sv} respectively such that $ru', rv' \in E_j$ (see Fig. 5 for an example). Without loss of generality, let $w(ru') \geq w(rv')$. Denote $P_{su'}$ and $P_{sv'}$ as the s-u' and s-v' subpaths of P_{su} and P_{sv} respectively. Now, consider the cycle $D := P_{su'} + P_{sv'} + ru' + rv'$. Observe that it contains r and is no bigger than C. Furthermore, it does not contain a chord incident to r by our choice of u' and v'. To arrive at a contradiction, it is left to show that D is well-covered, as this would imply D is violated. Suppose for a contradiction, that D is not well-covered. Then, there exists a chord g in D such that $w(g) < w(ru')$ and $w(g) < w(h)$ for some $h \in E(D)$, where ru' and h lie in different subcycles of D formed by g. This chord exists because C is well-covered but D is not. Let C_1 and C_2 denote the subcycles of C formed by g, where $h \in E(C_2)$. Note that any chord of C_1 covers C_1 because C is well-covered and $w(g) < w(h)$. Hence, the subcycle C_1 is well-covered. Moreover, we also have $w(g) \geq w(e)$ for all $e \in E(C_1)$ because C is well-covered and $w(g) < w(h)$. Let $w(g) = w_\ell$ for some $\ell < j$. Then, C_1 is still present in G_ℓ but ru' is not. Since C_1 also contains u and v but $uv \notin E_\ell$ because $uv \notin E_j$, it is a violated cycle in G_ℓ. However, this is a contradiction because there are no violated cycles in G_ℓ. □

The remaining proof proceeds in a similar fashion to the first case. Let $u, v \in V(C)$ be such that $uv \notin E_j$. By the claim above, we know that $ru \notin E_j$ or $rv \notin E_j$. For any chord f in C, observe that u and v lie in different subcycles of C formed by f. This is because the subcycles are well-covered by Claim 1, so the one which contains both u and v will contradict the minimality of C. Now, let g be a chord of C. Let C_u and C_v denote the subcycles of C formed by g where $u \in V(C_u)$ and $v \in V(C_v)$. The vertices of C_u are pairwise adjacent due to the minimality of C. Thus, C_u is a triangle. Otherwise, there exists a chord in C_u incident to u, and it forms a subcycle of C which contains both u and v. By an analogous argument, C_v is also a triangle. Therefore, C is a bad induced diamond in G_j. □

Notice that we have proven something stronger. Namely, if G_j contains a violated cycle, then there exists an index $i \leq j$ such that G_i contains a bad hole or a bad induced diamond. Moreover, as mentioned earlier, bad holes and Hamiltonian cycles of bad induced diamonds are violated cycles themselves. Thus, we obtain the following corollary.

Corollary 1 *There are no bad holes or bad induced diamonds in G_i for any $i < k$ if and only if there are no violated cycles in G_i for any $i < k$.*

4 Candidate edges and expensive neighborhood

In the previous section, we showed that violated cycles are an obstruction to submodularity. In light of this fact, we now focus on graphs G whose subgraphs G_i do not contain violated cycles. The goal of this section is to study the behaviour of f_{uv} on these graphs. As a first step, the following lemma sheds light on how a minimum spanning tree changes under vertex removal.

Lemma 4 *Assume that there are no violated cycles in G_i for any $i < k$. Let T be a minimum spanning tree of $G[S]$ where $r \in S \subseteq V$. For any $s \neq r$, there exists a minimum spanning tree of $G[S \setminus s]$ which contains $E(T \setminus s)$ and additionally, only uses edges from $G[N_T(s) \cup r]$.*

Proof Pick a vertex $s \in S \setminus r$. By Lemma 1, there exists a minimum spanning tree of $G[S \setminus s]$ which contains $E(T \setminus s)$. Let T' be such a tree which uses the most edges from $G[N_T(s) \cup r]$. We will show that T' is our desired tree. For the purpose of contradiction, suppose T' has an edge uv such that $uv \notin E(T)$ and $u \notin N_T(s) \cup r$. Note that u and v lie in different components of $T \setminus s$. Let P_{su} and P_{sv} denote the unique s-u and s-v paths in T respectively. We claim that $C := P_{su} + P_{sv} + uv$ is a well-covered cycle in G_i, where $w_i = w(uv)$. First, observe that uv is the most expensive edge in C because $P_{su} + P_{sv}$ is part of the minimum spanning tree T. Hence, the cycle C is present in G_i. Let $f = xy$ be any chord of C in G_i, and denote P_{xy} as the unique x-y path in T. Then, $P_{xy} + f$ is a subcycle of C formed by f. Moreover, we have $w(f) \geq w(e)$ for all $e \in E(P_{xy})$ because P_{xy} is part of the minimum spanning tree T. Hence, the chord f covers C, which proves the claim.

Let u' and v' be the vertices adjacent to s in P_{su} and P_{sv} respectively. By our choice of T', we have $w(u'v') > w(uv) = w_i$, which implies that u' and v' are not adjacent in G_i. Since there are no violated cycles in G_i, it follows that the vertices of C are adjacent to r in G_i. However, note that adding ru' or rv' to T' creates a fundamental cycle which uses the edge uv. Swapping it with uv creates another minimum spanning tree of $G[S \setminus s]$ which contains $E(T \setminus s)$ and uses more edges from $G[N_T(s) \cup r]$. We have arrived at a contradiction. $\square$

Given a pair of vertices $u, v \in N$ where $w(uv) = w_i$, the following definition distinguishes the neighbours of u, v in G from the neighbours of u, v in G_i.

Definition 4 For an edge $uv \in E$, if $w(uv) = w_i$, the *expensive neighborhood* of uv is defined as

$$\hat{N}(uv) := N_k(uv) \setminus N_i(uv).$$

In other words, the expensive neighborhood of an edge uv is the set of vertices $s \notin \{u, v\}$ such that $\max \{w(su), w(sv)\} > w(uv)$. It turns out that the function f_{uv} always returns zero when evaluated on a set which does not lie entirely in the expensive neighborhood of uv.

Lemma 5 *Assume that there are no violated cycles in G_i for any $i < k$. Let $u, v \in N$ and $S \in \mathcal{S}_{uv}$. If $S \nsubseteq \hat{N}(uv)$, then $f_{uv}(S) = 0$.*

Fig. 6 The left image depicts an example of the minimum spanning tree T in $G[S \cup \{u, v\}]$. The right image depicts an example of the minimum spanning tree T' in $G[S \cup v]$. The solid edges belong to the trees while dashed edges belong to the edge set F

Proof Let T be a minimum spanning tree of $G[S \cup \{u, v\}]$. First, we show that we can assume $uv \notin E(T)$. Since $S \nsubseteq \hat{N}(uv)$, there exists a vertex $s \in S$ such that $\max \{w(su), w(sv)\} \leq w(uv)$. If $uv \in E(T)$, then by rooting T at s, u is either a child or a parent of v. Adding su to T in the former and sv in the latter creates a fundamental cycle which contains uv. Thus, we can replace uv with this new edge to obtain the desired tree.

Since there are no violated cycles in G_i for any $i < k$, by Lemma 4, there exists a minimum spanning tree T' of $G[S \cup v]$ which contains $E(T \backslash u)$ and additionally, only uses edges from $G[N_T(u) \cup r]$. Recall that we assumed $uv \notin E(T)$, or equivalently, $v \notin N_T(u)$. Therefore, the neighborhood of v is identical in both trees, i.e. $N_T(v) = N_{T'}(v)$.

Consider the forest $T \backslash v$. Let $p \in N_T(v)$ such that p and r lie in the same component of $T \backslash v$ (see Fig. 6 for an example). Note that $p = r$ if $r \in N_T(v)$. We claim that p and r also lie in the same component of the forest $T' \backslash v$. We may assume that $p \neq r$, as otherwise the claim is trivially true. Moreover, we may assume that u lies on the unique p-r path in T. Otherwise, we are done because the same path is present in $T' \backslash v$. Let C_r denote the component of $T \backslash v$ which contains p, r and u. By Lemma 4, the endpoints of every edge in $E(T') \backslash E(T \backslash u)$ lie in C_r. This proves the claim.

Using Lemma 4, we can construct a minimum spanning tree of $G[S \cup u]$ by deleting v from T and adding a set F of edges from $G[N_T(v) \cup r]$. Note that $pr \notin F$ as p and r lie in the same component of $T \backslash v$. Since p and r also lie in the same component of $T' \backslash v$ and $N_T(v) = N_{T'}(v)$, deleting v from T' and adding the same set F of edges produces a minimum spanning tree of $G[S]$. Thus, we get

$$
\begin{aligned}
f_{uv}(S) &= \mathsf{mst}(S \cup u) + \mathsf{mst}(S \cup v) - \mathsf{mst}(S) - \mathsf{mst}(S \cup \{u, v\}) \\
&= \Big(\mathsf{mst}(S \cup u) - w(E(T))\Big) - \Big(\mathsf{mst}(S) - w(E(T'))\Big) \\
&= \Big(w(F) - w(\delta_T(v))\Big) - \Big(w(F) - w(\delta_{T'}(v))\Big) = 0
\end{aligned}
$$

as desired. $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\square$

We can now focus solely on vertex sets which lie entirely in the expensive neighborhood of uv. Observe that if $r \notin \hat{N}(uv)$, then $S \nsubseteq \hat{N}(uv)$ for all $S \in \mathcal{S}_{uv}$. Thus, we

do not have to check these edges uv as $f_{uv}(S) = 0$ for all $S \in \mathcal{S}_{uv}$ by the previous lemma. This motivates the following definition.

Definition 5 An edge $uv \in E$ is called a *candidate edge* if $r \in \hat{N}(uv)$.

With a mild assumption, we can show that the function f_{uv} is inclusion-wise non-increasing in the expensive neighborhood of uv.

Lemma 6 *Assume that there are no violated cycles in G_i for any $i < k$. Let uv be a candidate edge and $S \in \mathcal{S}_{uv}$ such that $S \subseteq \hat{N}(uv)$. If $f_{xy}(\hat{N}(xy)) \geq 0$ for every candidate edge xy, then $f_{uv}(S) \leq f_{uv}(S\backslash s)$ for any $s \neq r$.*

Proof Pick a vertex $s \in S\backslash r$. Without loss of generality, assume $w(su) \geq w(sv)$. Then, $w(su) > w(uv)$ because $s \in \hat{N}(uv)$. However, these two inequalities also imply that $v \notin \hat{N}(su)$. It follows that the set $(S\backslash s) \cup v$ is not contained in the expensive neighborhood of su. By Lemma 5,

$$0 = f_{su}((S\backslash s) \cup v) = \mathsf{mst}(S \cup v) + \mathsf{mst}((S\backslash s) \cup \{u, v\})$$
$$-\mathsf{mst}((S\backslash s) \cup v) - \mathsf{mst}(S \cup \{u, v\}).$$

Rearranging yields

$$\mathsf{mst}(S \cup v) - \mathsf{mst}(S \cup \{u, v\}) = \mathsf{mst}((S\backslash s) \cup v) - \mathsf{mst}((S\backslash s) \cup \{u, v\}). \quad (1)$$

Since uv is a candidate edge, let $w(uv) = w_i$ for some $i < k$. We will proceed by induction on i. For the base case $i = k - 1$, we have $w_{k-1} = w(uv) < w(su) = w_k$. Since $\hat{N}(su) = \emptyset$, the set $S\backslash s$ is not contained in the expensive neighborhood of su because $r \in S\backslash s$. By Lemma 5,

$$0 = f_{su}(S\backslash s) = \mathsf{mst}(S) + \mathsf{mst}((S\backslash s) \cup u) - \mathsf{mst}(S\backslash s) - \mathsf{mst}(S \cup u).$$

Rearranging yields

$$\mathsf{mst}(S \cup u) - \mathsf{mst}(S) = \mathsf{mst}((S\backslash s) \cup u) - \mathsf{mst}(S\backslash s). \quad (2)$$

Adding (1) and (2) gives $f_{uv}(S) = f_{uv}(S\backslash s)$. Now, suppose the lemma is true for all $i \geq j$ for some $j < k$. For the inductive step, let $w(uv) = w_{j-1}$. We may assume that $S\backslash s \subseteq \hat{N}(su)$, as otherwise we obtain equality again. This implies that su is a candidate edge because $r \in S\backslash s$. Since $w(su) > w(uv) = w_{j-1}$, we get

$$0 \leq f_{su}(\hat{N}(su)) \leq f_{su}(S\backslash s)$$
$$= \mathsf{mst}(S) + \mathsf{mst}((S\backslash s) \cup u) - \mathsf{mst}(S\backslash s) - \mathsf{mst}(S \cup u)$$

where the first inequality is due to our assumption while the second inequality is due to the induction hypothesis. Then, rearranging and adding it to (1) yields $f_{uv}(S) \leq f_{uv}(S\backslash s)$ as desired. $\qquad\square$

5 Characterization of submodularity

We are finally ready to give an efficient characterization of submodular spanning tree games.

Theorem 1 *The spanning tree game on G is submodular if and only if the following two conditions are satisfied.*

(i) There are no violated cycles in G_i for any $i < k$.
(ii) For every candidate edge uv, $f_{uv}(\hat{N}(uv)) \geq 0$.

Furthermore, these conditions can be verified in polynomial time.

Proof For necessity, assume the game is submodular. Then, Condition (i) is satisfied by Lemma 3 while Condition (ii) is satisfied trivially. For sufficiency, assume Conditions (i) and (ii) hold. Let $u, v \in N$ and $S \in \mathcal{S}_{uv}$. If $S \nsubseteq \hat{N}(uv)$, then $f_{uv}(S) = 0$ by Lemma 5. On the other hand, if $S \subseteq \hat{N}(uv)$, then uv is a candidate edge. By Lemma 6,

$$f_{uv}(S) \geq f_{uv}(\hat{N}(uv)) \geq 0.$$

Therefore, the game is submodular.

Checking Condition (ii) can clearly be done in polynomial time. Using Corollary 1, verifying Condition (i) reduces to searching for bad holes and bad induced diamonds in each G_i, which can be done efficiently. To look for bad holes, one could check if there exists a hole through a given vertex v for all $v \in N$ where $rv \notin E_i$. To look for bad induced diamonds, a naive implementation would involve examining all vertex subsets of size 4, which still runs in polynomial time. $\qquad\square$
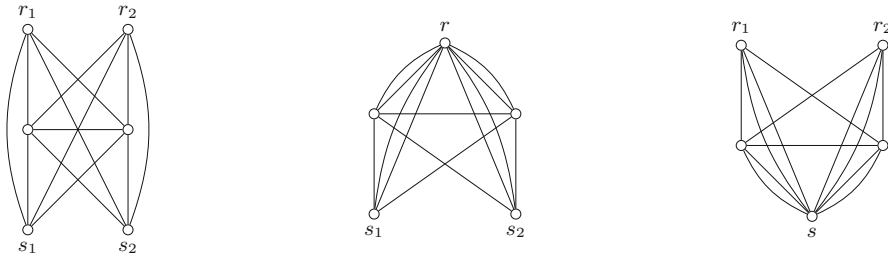
6 *S*-wide spanning trees

In this section, we answer another question posed in [9]. There the authors stated a necessary condition for submodularity of the spanning tree game, and left open whether their condition can be verified in polynomial time. We here show that this is indeed the case. While this is not that relevant anymore in order to characterize submodularity (since we have provided Theorem 1), it shows a nice connection with matroid intersection that might still be of interest.

Theorem 2 (Theorem 1.2 in [9]). *If the spanning tree game on G is submodular, then every minimum spanning tree T of G possesses the following two properties.*

(a) It holds that $w(rv) \geq w(ru)$ for every vertex $v \in N$ and every vertex $u \in N$ on the (unique) path connecting r and v in T.
(b) For any pair of vertices $u, v \in N$ where $w(uv) < w(rv)$, the cycle obtained by adding uv to T does not contain r.

We show that checking Conditions (a) and (b) can be reduced to computing *S-wide* spanning trees, which are defined as follows. Let $G = (V, E)$ be an undirected graph with edge weights $w : E \to \mathbb{R}$ and a designated root $r_1 \in V$. Let $S := \{s_1, \ldots, s_k\} \subseteq$

Fig. 7 The graph G' and multigraphs G'_1, G'_2 respectively for the input graph K_5 with 2 terminals

$V \backslash r_1$ be a given set of *terminals*. We say that a spanning tree T of G is S-*wide* if every component of $T \backslash r_1$ contains at most one terminal. Equivalently, T is S-*wide* if for every $i, j \in [k]$ where $i \neq j$, the unique s_i-s_j path on T contains the root r_1. We are interested in the following problem:

Given (G, w, r_1, S), compute an S-wide spanning tree of minimum weight.

Before solving the problem above, let us explain how one can use it to efficiently test Conditions (a) and (b) of Theorem 2. To check Condition (a), consider all pairs of vertices $u, v \in N$ with $w(rv) < w(ru)$, and do the following. Set $r_1 := u$ and $S := \{v, r\}$. Compute an S-wide spanning tree T^* of minimum weight in G. If the weight of T^* is equal to the weight of a minimum spanning tree in G, then T^* is a minimum spanning tree of G violating Condition (a), since u is on the unique path from r to v in T^*. It is not difficult to see that this procedure will eventually find a minimum spanning tree violating Condition (a), if one exists.

Condition (b) can be checked in a similar way. Consider all ordered pairs of vertices $u, v \in N$ with $w(uv) < w(rv)$, and do the following. Set $r_1 := r$ and $S := \{u, v\}$. Compute an S-wide spanning tree T^* of minimum weight in G. If the weight of T^* is equal to the weight of a minimum spanning tree in G, then T^* is a minimum spanning tree of G violating Condition (b), since adding uv to T^* yields a cycle containing r. Once again, it is not difficult to see that this procedure will eventually find a minimum spanning tree violating Condition (b), if one exists.

We will now demonstrate how to compute an S-wide spanning tree of minimum weight using matroid intersection. Let (G, w, r_1, S) be a given instance. Without loss of generality, we may assume that there are no edges between any pair of terminals, as every S-wide tree does not use them. We also assume $k \geq 2$, otherwise this reduces trivially to computing an arbitrary minimum spanning tree.

First, construct an auxiliary graph $G' = (V', E')$ from G as follows. Create $k - 1$ copies of the root vertex $r_2, \ldots, r_k$, including its incident edges. Next, let G'_1 denote the multigraph obtained from G' by identifying $r_1, \ldots, r_k$ into a single vertex r. Note that parallel edges are kept. In order to distinguish the k parallel edges between r and some vertex v, they will be denoted $r_i v$ for all $i \in [k]$, just like in the graph G'. Similarly, let G'_2 denote the multigraph obtained from G' by identifying $s_1, \ldots, s_k$ into a single vertex s. See Fig. 7 for an example.

Now, consider the graphic matroids on G_1' and G_2', denoted $M_1 = (E', \mathcal{I}_1)$ and $M_2 = (E', \mathcal{I}_2)$ respectively. Observe that the matroids M_1 and M_2 share the same ground set E' due to our notation of the parallel edges, but their independent sets are different. We would like to establish a correspondence between S-wide spanning trees in G and common bases of M_1 and M_2. Let $\mathcal{T}$ be the set of S-wide spanning trees in G. Moreover, let $\mathcal{B}_1$ and $\mathcal{B}_2$ be the set of bases of M_1 and M_2 respectively. Construct the function $g : 2^{E'} \to 2^E$ defined by

$$g(J) := \{h(e) : e \in J\},$$

where $h : E' \to E$ is defined as

$$h(uv) := \begin{cases} r_1 v, & \text{if } u \in \{r_1, \ldots, r_k\} \\ u r_1, & \text{if } v \in \{r_1, \ldots, r_k\} \\ uv, & \text{otherwise.} \end{cases}$$

Lemma 7 *For every $J \in \mathcal{B}_1 \cap \mathcal{B}_2$, $g(J)$ induces an S-wide spanning tree in G.*

Proof Let $J \in \mathcal{B}_1 \cap \mathcal{B}_2$. Then, it induces a spanning tree in the multigraphs G_1' and G_2'. Note that $g(J)$ also induces a spanning tree in the multigraph G_1'. Since the input graph G can be obtained from the multigraph G_1' by removing parallel edges, and $g(J) \subseteq E$, it follows that $g(J)$ induces a spanning tree T in G. It is left to show that T is S-wide. For the purpose of contradiction, suppose there exist distinct indices $i, j \in [k]$ such that the unique s_i-s_j path P in T does not contain the root r_1. From the definition of h, we know that $E(P) \subseteq J$. However, $E(P)$ induces a non-simple walk in G_2', which implies that $J \notin \mathcal{I}_2$. We have reached a contradiction. □

Lemma 8 *For every $T \in \mathcal{T}$, there exists a set $J \in \mathcal{B}_1 \cap \mathcal{B}_2$ such that $g(J) = E(T)$.*

Proof Let T be an S-wide spanning tree in G. For every $i \in [k]$, let $r_1 v_i$ be the first edge of the r_1-s_i path on T. Replace each $r_1 v_i$ with $r_i v_i$ and call the resulting edge set J. Then, J induces a forest F with k components in the auxiliary graph G', each of which contains r_i and s_i. Since $\delta_F(v) \geq 1$ for all $v \in V'$, J induces a spanning tree in the multigraph G_1', as well as a spanning tree in the multigraph G_2'. Thus, $J \in \mathcal{B}_1 \cap \mathcal{B}_2$. Moreover, $g(J) = E(T)$. □

The last two lemmas imply that g is a surjective mapping from $\mathcal{B}_1 \cap \mathcal{B}_2$ to $\mathcal{T}$. It is also cost-preserving because $|J| = |g(J)|$ for all $J \in \mathcal{B}_1$. Therefore, we can efficiently compute a minimum-weight S-wide spanning tree by computing a minimum-weight common basis of M_1 and M_2.

References

1. Bird, C.G.: On cost allocation for a spanning tree: a game theoretic approach. Networks **6**(4), 335–350 (1976)
2. Chalkiadakis, G., Elkind, E., Wooldridge, M.: Computational Aspects of Cooperative Game Theory. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers (2011)
3. Claus, A., Kleitman, D.J.: Cost allocation for a spanning tree. Networks **3**(4), 289–304 (1973)
4. Faigle, U., Kern, W., Fekete, S.P., Hochstättler, W.: On the complexity of testing membership in the core of min-cost spanning tree games. Int. J. Game Theory **26**(3), 361–366 (1997)
5. Faigle, U., Kern, W., Kuipers, J.: Note computing the nucleolus of min-cost spanning tree games is NP-hard. Int. J. Game Theory **27**(3), 443–450 (1998)
6. Granot, D., Huberman, G.: Minimum cost spanning tree games. Math. Program. **21**(1), 1–18 (1981)
7. Granot, D., Huberman, G.: The relationship between convex games and minimum cost spanning tree games: a case for permutationally convex games. SIAM J. Algebr. Dis. Methods **3**(3), 288–292 (1982)
8. Grötschel, M., Lovász, L., Schrijver, A.: Geometric Algorithms and Combinatorial Optimization, Volume 2 of Algorithms and Combinatorics. Springer, Berlin (1993)
9. Kobayashi, M., Okamoto, Y.: Submodularity of minimum-cost spanning tree games. Networks **63**(3), 231–238 (2014)
10. Kuipers, J.: A polynomial time algorithm for computing the nucleolus of convex games. Report M, (Maastricht University, 1996). pp. 96–12
11. Maschler, M., Peleg, B., Shapley, L.S.: The kernel and bargaining set for convex games. Int. J. Game Theory **1**(1), 73–93 (1971)
12. Okamoto, Y.: Submodularity of some classes of the combinatorial optimization games. Math. Methods Oper. Res. **58**(1), 131–139 (2003)
13. Shapley, L.S.: Cores of convex games. Int. J. Game Theory **1**(1), 11–26 (1971)
14. Trudeau, C.: A new stable and more responsive cost sharing solution for minimum cost spanning tree problems. Games Econ. Behav. **75**(1), 402–412 (2012)
15. van den Nouweland, A., Borm, P.: On the convexity of communication games. Int. J. Game Theory **19**(4), 421–430 (1991)