Submodularity of Minimum-Cost Spanning Tree Games

Masayuki Kobayashi* Yoshio Okamoto†

Abstract

We give a necessary condition and a sufficient condition for a minimum-cost spanning tree game introduced by Bird to be submodular (or convex). When the cost is restricted to two values, we give a characterization of submodular minimum-cost spanning tree games. We also discuss algorithmic issues.
Keywords: Convex game; Cooperative game; Cost allocation; Minimum-cost spanning tree game; Submodularity

1 Introduction

Some cost allocation problems can be modeled within the framework of cooperative game theory. In particular, we will study the following problem.

A service will be provided by a central server, and there are several customers who wish to enjoy the service. We want to construct a network so that every customer can enjoy the service and the construction cost is as cheap as possible. This can be modeled as the minimum-cost spanning tree problem, which can be solved efficiently. Now the cost allocation comes into play: We want to allocate (or distribute) the minimized total cost to each customer so that the allocation can be seen as "fair" in a certain sense.

We model the problem more formally in the following way. Let N be a finite set corresponding to the set of customers and $s \notin N$ be a server. We consider a complete graph $G = (N \cup \{s\}, E)$ where E consists of all unordered pairs of $N \cup \{s\}$. In addition, let $w \colon E \to \mathbb{R}_+$ be a nonnegative function on the set of edges of G which represents the cost needed for the construction of a link between two vertices in G. Note that we have no assumption on w; for example w does not have to satisfy the triangle inequality. A *spanning tree* in G is a subgraph $T = (V_T, E_T)$ of G such that T is a tree (i.e., a connected graph without cycles) and $V_T = V$. A *minimum-cost spanning tree* of G is a spanning tree T such that the sum of the weights $w(e)$ of all edges e in T is the smallest among all spanning trees in G. Then the *minimum-cost spanning tree game* arising from G and w is a pair (N, mcst) of the set N called the *player set* and a function $\mathsf{mcst} \colon 2^N \to \mathbb{R}$, called the *characteristic function* of the game, defined so that $\mathsf{mcst}(S)$ is the cost of a minimum-cost spanning tree in $G[S \cup \{s\}]$. Here $G[S \cup \{s\}]$ is the subgraph of G induced by $S \cup \{s\}$. One of the goals of cooperative game theory is to allocate the total cost (in our case $\mathsf{mcst}(N)$) to each player (in our case each element of N) in a "fair" manner. The minimum-cost spanning tree game as described above was first introduced by Bird [1].

*Department of Information and Computer Sciences, Toyohashi University of Technology, Hibarigaoka 1–1, Tempaku, Toyohashi, Aichi, 441–8580, Japan.

†Corresponding author. Department of Communication Engineering and Informatics, Graduate School of Informatics and Engineering, The University of Electro-Communications, Chofugaoka 1–1, Chofu, Tokyo, 182-8585, Japan. Supported by Grant-in-Aid for Scientific Research from Ministry of Education, Science and Culture, Japan, and Japan Society for the Promotion of Science (JSPS).

The aim of this work is to try to characterize minimum-cost spanning tree games with nice properties. In particular, we concentrate on submodularity. A minimum-cost spanning tree game (N, mcst) is *submodular* (or *convex*) if for every $S, T \subseteq N$ it holds that $\mathsf{mcst}(S)+\mathsf{mcst}(T) \geq \mathsf{mcst}(S \cap T)+\mathsf{mcst}(S \cup T)$. It is known that a submodular game possesses several good properties: The core is always non-empty and it is a unique von Neumann-Morgenstern solution [14]; The Shapley value is the barycenter of the vertices of the core (when the degeneracy is taken into account) [14]; The core and the bargaining set coincide and the kernel and the nucleolus coincide [10]; The nucleolus and the τ-value can be computed in polynomial time ([8] and [16] respectively).

Unfortunately, we have not succeeded in providing a characterization of submodular minimum-cost spanning games. We feel that recognizing a submodular minimum-cost spanning tree game is coNP-complete (implying that no good characterization is likely to exist), but we are still far from proving such a result.

Therefore, we concentrate on a sufficient condition and a necessary condition. First we show a sufficient condition for submodularity.

Theorem 1.1. *A minimum-cost spanning tree game* (N, mcst) *is submodular if every minimum-cost spanning tree* T *of* G *possesses the following two properties.*

1. *It holds that* $w(s, v) \geq w(s, u)$ *for every vertex* $v \in N$ *and every vertex* $u \in N$ *on the (unique) path connecting* s *and* v *in* T*.*

2. *For every edge* $\{u, v\} \notin E(T)$*, it holds that* $w(u, v) \geq w(s, v)$*.*

Furthermore, this condition can be verified in polynomial time.

The next theorem gives a necessary condition for submodularity.

Theorem 1.2. *If a minimum-cost spanning tree game* (N, mcst) *is submodular, then every minimum-cost spanning tree* T *of* G *possesses. the following two properties.*

1. *It holds that* $w(s, v) \geq w(s, u)$ *for every vertex* $v \in N$ *and every vertex* $u \in N$ *on the (unique) path connecting* s *and* v *in* T*.*

2. *For any pair of vertices* $u, v \in N$ *such that* $w(u, v) < w(s, v)$ *the subgraph* $T + \{u, v\}$ *does not contain a cycle through* s*.*

We do not know whether the condition in Theorem 1.2 can be verified in polynomial time.

Next, we consider the case in which the edge weights are bound to two values $w \colon E \to \{a, b\}$ where $a < b$. This case occurs when the valuation of the weights is done by binary decision. For example, a computer network may have two setup costs, high and low, depending on the required security of the link, and the designer decides which cost is assigned to each link.

For this case, we are able to give a characterization for submodularity.

Theorem 1.3. *When the edge weights are restricted to two values, i.e.,* $w \colon E \to \{a, b\}$ *where* $a < b$*, a minimum-cost spanning tree game* (N, mcst) *is submodular if and only if for the graph* G' *obtained by deleting all edges of weight* b*, the vertices of every cycle* C *in* G' *are adjacent to* s *or they are pairwise adjacent. Furthermore, this condition can be verified in polynomial time.*

Related work A cost allocation problem arising from the minimum-cost spanning tree problem was first introduced by Claus and Kleitman [2]. Bird [1] gave a game-theoretic perspective to this problem and proposed the so-called Bird's allocation, which was proved to belong to the core by Granot and Huberman [5]. Since then, various aspects of minimum-cost spanning tree games have been investigated (we omit the vast literature). Among them, Granot and Huberman [6] proved that a minimum-cost spanning tree game is permutationally convex (which is a generalization of submodularity). However, no characterization has been given for submodular minimum-cost spanning tree games. Even little is known about submodular (or convex) combinatorial optimization games in spite of the vast literature on the topic (e.g. see Curiel's book [3]). A related submodularity result has been provided by Trudeau [17], which gave a sufficient condition for the stand-alone game arising from a minimum-cost spanning tree game to be submodular. Another partial result has been provided by the second author [12] who characterized submodular minimum coloring games and submodular minimum vertex cover games introduced by Deng, Ibaraki, and Nagamochi [4].

Several other models are known for cost allocation in network design such as Steiner tree games [11], minimum-cost spanning forest games [13], and minimum-cost forest games [9]. The underlying optimization problems of these games are NP-hard. Therefore, the structural properties of these games, including submodularity, are unlikely to be similar to that of minimum-cost spanning tree games.

2 Preliminaries

Notation For a graph $G = (V, E)$ and a pair of vertices $u, v \in V$, $G + \{u, v\}$ means the graph obtained from G by adding the edge $\{u, v\}$, and $G - \{u, v\}$ means the graph obtained from G by deleting the edge $\{u, v\}$. This notation is extended to a set of edges as follows. If F is a set of pairs of vertices of G, then $G + F$ means the graph obtained from G by adding the edges in F, and $G - F$ means the graph obtained from G by deleting the edges in F. For a vertex subset $S \subseteq V$, we denote by $G[S]$ the subgraph of G induced by S. Namely, the vertex set of $G[S]$ is S, and the edge set of $G[S]$ consists of the edges of G that have both endpoints in S.

Rooted trees In this paper, the vertex s is treated as a special vertex. Whenever we talk about a tree T containing s, we regard it as a tree rooted at s. Then, we apply usual terms for rooted trees. For example, a vertex v is an *ancestor* of a vertex u if v lies on a unique path from v to s. Furthermore, if u and v are adjacent in T, then v is the *parent* of u, and u is a *child* of v.

Algorithms for minimum-cost spanning trees In the course of proofs, we often construct a minimum-cost spanning tree of a graph, and we assume how we can find one. Here, we present two famous algorithms, which will be used explicitly or implicitly.

The first algorithm is a greedy algorithm due to Kruskal [7], which works as follows. Let $G = (V, E)$ be a connected graph with non-negative edge cost function $w : E \to \mathbb{R}_+$. We first sort the edges of G as $e_1, e_2, \ldots, e_m$ such that $w(e_1) \leq w(e_2) \leq \cdots \leq w(e_m)$. For each $i = 1, \ldots, m$, we define $F_i \subseteq E$ as follows. When $i = 1$, $F_i = \{e_1\}$. When $i > 1$, $F_i = F_{i-1} \cup \{e_i\}$ if $F_{i-1} \cup \{e_i\}$ contains no cycle; otherwise $F_i = F_{i-1}$. Then, we can prove that F_m is the edge set of a minimum-cost spanning tree of G.

The following property can be proved directly or by means of Kruskal's algorithm. Let T be a minimum-cost spanning tree of $G = (V, E)$ and $S \subseteq V$. If $T[S]$ is connected, then $T[S]$ is a minimum-cost spanning tree of $G[S]$.

The second algorithm is based on local improvement. Let T be a spanning tree of the graph $G = (V, E)$. For every edge $e \in E \setminus E(T)$, there exists a unique cycle C in $T + e$. Note that C contains e. Then, for every edge $e' \in E(C) \setminus \{e\}$, $T' = (T + e) - e'$ is a spanning tree of G. If $w(e) < w(e')$, then the cost of T' is smaller than that of T. If we repeat this procedure

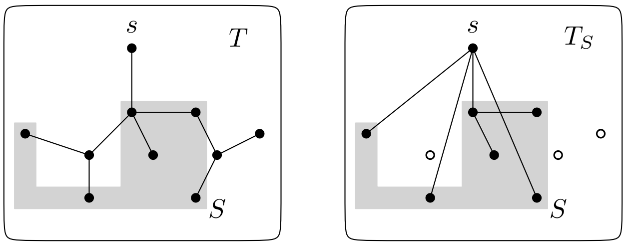

Figure 1: The construction of T_S in Lemma 3.1.

to obtain a spanning tree of smaller cost until no such pair of edges exists, then the result is a minimum-cost spanning tree of G. Note also that this always terminates.

3 Proof of Theorem 1.1

To prove the first half of Theorem 1.1, we use the following lemma.

Lemma 3.1. *Let T be an arbitrary minimum-cost spanning tree of $G = (N \cup \{s\}, E)$. We construct a spanning tree T_S of $G[S \cup \{s\}]$ as follows. For each $v \in S$, let $p(v) \in N \cup \{s\}$ be the parent of v in T. Then we set $E(T_S) = \{\{v, p(v)\} \mid v \in S, p(v) \in S\} \cup \{\{s, v\} \mid v \in S, p(v) \notin S\}$.*

Then, T_S is a minimum-cost spanning tree of $G[S \cup \{s\}]$ if the following two conditions are satisfied.

1. *It holds that $w(s, v) \geq w(s, u)$ for every vertex $v \in N$ and every ancestor $u \in N$ of v in T.*

2. *It holds that $w(s, v) \leq w(u, v)$ for every $\{u, v\} \notin T$.*

Figure 1 illustrates the construction of T_S. Note that the two conditions in Lemma 3.1 are identical to those in Theorem 1.1.

Proof. It is not difficult to see that the constructed graph T_S is indeed a spanning tree of $G[S \cup \{s\}]$. Now, we claim that T_S is a minimum-cost spanning tree of $G[S \cup \{s\}]$.

Let T'_S be a minimum-cost spanning tree of $G[S \cup \{s\}]$. We proceed by induction on $|E(T'_S) \setminus E(T_S)|$. If $|E(T'_S) \setminus E(T_S)| = 0$, then $T'_S = T_S$ and we are done. For the induction step, we assume $|E(T'_S) \setminus E(T_S)| > 0$. Then, there must be $v \in S$ with $p(v) \in S$ such that $\{v, p(v)\} \notin E(T'_S)$ or $v \in S$ with $p(v) \notin S$ such that $\{s, v\} \notin E(T'_S)$. Therefore, we have two cases.

For the first case, assume the existence of $v \in S$ with $p(v) \in S$ such that $\{v, p(v)\} \notin E(T'_S)$. Then $T'_S + \{v, p(v)\}$ contains a cycle C in $G[S \cup \{s\}]$. Since T is a minimum-cost spanning tree of G and $\{v, p(v)\} \in E(T)$, there must exist an edge e of C that is not an edge of T. Note that $w(e) \geq w(\{v, p(v)\})$ since $T + e - \{v, p(v)\}$ is a spanning tree of G and T is a minimum-cost spanning tree of G. On the other hand, $w(e) \leq w(\{v, p(v)\})$ since $T'_S + \{v, p(v)\} - e$ is a spanning tree of $G[S \cup \{s\}]$ and T'_S is a minimum-cost spanning tree of that graph. Consequently we have $w(e) = w(\{v, p(v)\})$, and so $T'_S + \{v, p(v)\} - e$ is also a minimum-cost spanning tree of $G[S \cup \{s\}]$. Since $|E(T'_S + \{v, p(v)\} - e) \setminus E(T_S)| = |E(T'_S) \setminus E(T_S)| - 1$, the claim holds by the induction hypothesis.

For the second case, assume the existence of $v \in S$ with $p(v) \notin S$ such that $\{s, v\} \notin E(T'_S)$. Then, $T'_S + \{s, v\}$ contains a cycle C in $G[S \cup \{s\}]$. The edges of $C - \{s, v\}$ are those on the unique path P from v to s in T'_S. Since $p(v) \notin S$ and $p(v)$ belongs to the unique path from v to s in T by definition, there must exist an edge of P that is not an edge of T. Let x, y be the endpoints of the edge closest to v with such a property, and assume that x is closer to v

than y in P. Namely, the subpath of P from v to x is completely contained in T. Note that $x \neq s$ and $w(s,x)$ is well-defined. Then, by the second condition of the lemma, it follows that $w(x,y) \geq w(s,x)$. Now observe that v is an ancestor of x in T since $p(v)$ does not belong to P. Hence, by the first condition of the lemma, it holds that $w(s,x) \geq w(s,v)$. Namely, we see that $w(x,y) \geq w(s,v)$. Therefore, the cost of the spanning tree $T'_S + \{s,v\} - \{x,y\}$ of $G[S \cup \{s\}]$ is at most the cost of T'_S. This means that $T'_S + \{s,v\} - \{x,y\}$ is a minimum-cost spanning tree of $G[S \cup \{s\}]$, and furthermore, $|E(T'_S + \{s,v\} - \{x,y\}) \setminus E(T_S)| = |E(T'_S) \setminus E(T_S)| - 1$. By the induction hypothesis, the claim holds. □

We are ready to prove the first half of Theorem 1.1.

Proof of the first half of Theorem 1.1. Assume the conditions in the statement of Theorem 1.1 (and thus those in Lemma 3.1) hold. It is well-known and not hard to see that the submodularity is equivalent to the following condition: for every $i,j \in N$ and $S \subseteq N \setminus \{i,j\}$,

$$\mathsf{mcst}(S \cup \{i\}) + \mathsf{mcst}(S \cup \{j\}) \geq \mathsf{mcst}(S) + \mathsf{mcst}(S \cup \{i,j\}).$$

Let $i,j \in N$ and $S \subseteq N \setminus \{i,j\}$ be arbitrarily chosen. Further let T be an arbitrary minimum-cost spanning tree of G, and denote by $T_{S\cup\{i\}}$ the minimum-cost spanning tree of $G[S \cup \{i,s\}]$ constructed as in Lemma 3.1 from T.

Denote by U the set of children of i in T. Then, by the construction of $T_{S\cup\{i\}}$, every vertex of $U \cap S$ is adjacent to i in $T_{S\cup\{i\}}$. If we remove i from $T_{S\cup\{i\}}$ and make the vertices in $U \cap S$ adjacent to s, then the resulting graph is a minimum-cost spanning tree of $G[S \cup \{s\}]$ by Lemma 3.1. Therefore, if we denote by E_1 the set of edges incident to i in $T_{S\cup\{i\}}$, and by E_2 the set of edges connecting s and the vertices in $U \cap S$, then it follows that

$$\mathsf{mcst}(S) = \mathsf{mcst}(S \cup \{i\}) - \sum_{e \in E_1} w(e) + \sum_{e \in E_2} w(e). \tag{1}$$

Now, consider the minimum-cost spanning tree $T_{S\cup\{j\}}$ of $G[S \cup \{j,s\}]$ constructed as in Lemma 3.1 from T. Since E_2 is completely contained in $T_{S\cup\{j\}}$ by construction, we can see that $T_{S\cup\{j\}} - E_2 + E_1$ is a spanning tree of $G[S \cup \{i,j,s\}]$. Therefore, it follows that

$$\mathsf{mcst}(S \cup \{i,j\}) \leq \mathsf{mcst}(S \cup \{j\}) - \sum_{e \in E_2} w(e) + \sum_{e \in E_1} w(e). \tag{2}$$

Putting (1) and (2) together, we obtain $\mathsf{mcst}(S \cup \{i\}) + \mathsf{mcst}(S \cup \{j\}) \geq \mathsf{mcst}(S) + \mathsf{mcst}(S \cup \{i,j\})$. □

To prove the second half (the algorithmic part), we use the following lemma.

Lemma 3.2. *The following two are equivalent.*

1. *For* every *minimum-cost spanning tree T of G, it holds that $w(s,v) \geq w(s,u)$ for every vertex $v \in N$ and every ancestor $u \in N$ of v in T and for every edge $\{u,v\} \notin E(T)$ it holds that $w(u,v) \geq w(s,v)$.*

2. *For* some *minimum-cost spanning tree T of G, it holds that $w(s,v) \geq w(s,u)$ for every vertex $v \in N$ and every ancestor $u \in N$ of v in T and for every edge $\{u,v\} \notin E(T)$ it holds that $w(u,v) \geq w(s,v)$.*

Note that the first statement of the lemma is exactly the condition in Theorem 1.1. Hence, Lemma 3.2 immediately gives a polynomial-time algorithm: We just need to look at one arbitrary minimum-cost spanning tree of G. This will complete the proof of the second half of Theorem 1.1.

Proof of Lemma 3.2. The direction "1 $\Rightarrow$ 2" is trivial. We prove the other direction. Assume that Statement 2 holds. Let T be a minimum-cost spanning tree of G in Statement 2. It is well-known that every minimum-cost spanning tree of a graph can be obtained from another minimum-cost spanning tree of the graph by a sequence of the following operations: add one edge of minimum weight, which creates a cycle, and remove one edge of maximum weight in the cycle. Note that this operation does not increase the weight of the obtained spanning tree, and thus all the spanning trees obtained in this sequence of processes are minimum-cost spanning trees. Therefore, it is enough to show that when we apply this operation to T, the resulting spanning tree still satisfies the condition.

Consider a minimum-cost spanning tree $T' = T + \{u, v\} - \{p, r\}$ obtained from T by the operation above. Then, $w(u, v) = w(p, r)$ since T' and T are minimum-cost spanning trees. Observe that $\{p, r\}$ belongs to a unique cycle in $T + \{u, v\}$. Without loss of generality, we assume that $\{p, r\}$ lies on the path P from v to s in T, and p is the parent of r in T. This also implies that $s \neq v$ and $w(s, v)$ is well-defined. By the first condition in Statement 2, it holds that $w(s, v) \geq w(s, r) \geq w(s, p)$. Furthermore, it holds that $w(u, v) \geq w(s, v)$ by the second condition of Statement 2.

Consider the cycle $P + \{s, v\}$. Since $\{s, v\}$ does not belong to T but $\{p, r\}$ belongs to T (and P), it holds that $w(s, v) \geq w(p, r)$. Similarly, it holds that $w(s, r) \geq w(p, r)$. They imply that $w(s, r) \geq w(p, r) = w(u, v) \geq w(s, v) \geq w(s, r)$. Therefore it follows that $w(s, r) = w(p, r) = w(u, v) = w(s, v)$. Then, by the first condition in Statement 2, for every vertex x on the path connecting v and r in T it holds that $w(s, r) = w(s, x) = w(s, v)$.

We are ready for verifying that T' satisfies the conditions. For the first condition, choose any vertex $x \in N$. If the paths from x to s are identical in T and T', we are done. Otherwise, x must lie on the path connecting v and r in T. However, for such x we already saw that $w(s, r) = w(s, x) = w(s, v)$. Therefore, the first condition holds. For the second condition, we only need to look at the edge $\{p, r\}$ because this is the only non-edge of T' that was an edge of T. However, we have already seen that $w(p, r) = w(s, r) \geq w(s, p)$. This completes the proof. $\square$

4 Proof of Theorem 1.2

Proof of Theorem 1.2. We will show the contrapositive: a minimum-cost spanning tree game (N, mcst) is not submodular if for some minimum-cost spanning tree T of G, one of the following two properties is satisfied.

1. There exist a vertex $v \in N$ and an ancestor $u \in N$ of v in T such that $w(s, v) < w(s, u)$.

2. There exist vertices $u, v \in N$ such that $w(u, v) < w(s, v)$ and the subgraph $T + \{u, v\}$ contains a cycle through s.

Suppose that a minimum-cost spanning tree T of G satisfies the condition above. We have two cases. Assume first that there exist a vertex $v \in N$ and an ancestor $u \in N$ of v in T such that $w(s, v) < w(s, u)$. Without loss of generality, we may assume that u is next to v, and let v be the closest one from s in T among such vertices. Note that u is not next to s in T since otherwise $T + \{s, v\} - \{s, u\}$ would be a spanning tree with cost smaller than T. Let P be a maximal subpath of the path connecting u and s containing u and on which every vertex x satisfies $w(s, x) = w(s, u)$. We denote by r the endpoint of P that lies between u and s. Note that r is identical to u when the length of P is zero. Then, we see that there exists a vertex on the path from r to s in T that is not r or s since otherwise s and r are adjacent in T, but a spanning tree $T + \{s, v\} - \{s, r\}$ has a smaller cost than T, which contradicts the minimality of T. Let p be the vertex on the path from r to s in T that is next to r. Then, it holds that $w(s,p) < w(s,r)$ by the choice of v, r, and P.

Now, we observe that

$$\begin{aligned}
\mathsf{mcst}(V(P)) &= \sum_{e \in E(P)} w(e) + w(s,u),\\
\mathsf{mcst}(V(P) \cup \{v\}) &= \sum_{e \in E(P)} w(e) + w(s,v) + w(u,v),\\
\mathsf{mcst}(V(P) \cup \{p\}) &= \sum_{e \in E(P)} w(e) + w(s,p) + w(r,p),\\
\mathsf{mcst}(V(P) \cup \{p,v\}) &= \sum_{e \in E(P)} w(e) + w(u,v) + w(r,p) + \min\{w(s,v), w(s,p)\}
\end{aligned}$$

since T is a minimum-cost spanning tree of G and P is completely contained in T. Hence, we have

$$\begin{aligned}
&\mathsf{mcst}(V(P) \cup \{v\}) + \mathsf{mcst}(V(P) \cup \{p\}) - \mathsf{mcst}(V(P)) - \mathsf{mcst}(V(P) \cup \{p,v\})\\
&\quad = w(s,v) + w(s,p) - w(s,u) - \min\{w(s,v), w(s,p)\}\\
&\quad = \max\{w(s,v), w(s,p)\} - w(s,u)\\
&\quad < 0,
\end{aligned}$$

and so (N, mcst) is not submodular.

For the second case, assume that there exist vertices $u, v \in N$ such that $w(u,v) < w(s,v)$ and the subgraph $T + \{u,v\}$ contains a cycle C through s. Choose $u, v \in N$ such that the cycle C is the shortest.

Let P be a maximal subpath of the path in T connecting v and s which contains v and on which every vertex x satisfies $w(s,x) = w(s,v)$. Denote by r the endpoint of P that lies between v and s. Note that r can be idential to v when the length of P is zero. Then, there exists a vertex on a path from r to s in T that is not r or s since otherwise s and r are adjacent in T, but a spanning tree $T + \{u,v\} - \{s,r\}$ has a smaller cost than T, which contradicts the minimality of T. Let p be a vertex on the path from r to s in T that is next to r. Then, we may assume that $w(s,p) < w(s,r)$ since otherwise the situation is reduced to the first case.

Let P' be a unique path from u to s in T. By the choice of u, v, and since C is a shortest cycle, for every $x \in V(P' - \{s,u\})$ and every $y \in V(P)$, it holds that $w(x,y) \geq w(s,y) = w(s,v)$. Note also that P and P' have no common vertex since s belongs to a unique cycle in $T + \{u,v\}$. Thus, if we let $S = V(P) \cup V(P' - \{s,u\})$, then we have

$$\begin{aligned}
\mathsf{mcst}(S) &= \sum_{e \in E(P)} w(e) + \sum_{e \in E(P' - \{u\})} w(e) + w(s,v),\\
\mathsf{mcst}(S \cup \{u\}) &= \sum_{e \in E(P)} w(e) + \sum_{e \in E(P')} w(e) + w(u,v),\\
\mathsf{mcst}(S \cup \{p\}) &= \sum_{e \in E(P)} w(e) + \sum_{e \in E(P' - \{u\})} w(e) + w(s,p) + w(r,p),\\
\mathsf{mcst}(S \cup \{p,u\}) &= \sum_{e \in E(P)} w(e) + \sum_{e \in E(P')} w(e) + w(r,p) + \min\{w(u,v), w(s,p)\}.
\end{aligned}$$

Hence,

$$\begin{aligned}
&\mathsf{mcst}(S \cup \{u\}) + \mathsf{mcst}(S \cup \{p\}) - \mathsf{mcst}(S) - \mathsf{mcst}(S \cup \{p, u\}) \\
&\quad = w(u, v) + w(s, p) - w(s, v) - \min\{w(u, v), w(s, p)\} \\
&\quad = \max\{w(u, v), w(s, p)\} - w(s, v) \\
&\quad < 0,
\end{aligned}$$

and so (N, mcst) is not submodular. □

5 Proof of Theorem 1.3

First notice that the submodularity does not depend on the values a and b themselves, but it only depends on how the values are distributed over the edges. This is because any spanning tree of a graph with k vertices has $k-1$ edges, and so in the submodular inequality the total numbers of edges involved are identical on both sides.

5.1 Proof of the only-if part

Proof of the only-if part of Theorem 1.3. First we show the contrapositive of the only-if part. To this end, consider a minimal cycle C in G', namely $V(C)$ does not contain any cycle other than C. Then, the vertices of C are not pairwise adjacent, and there exists a vertex that is not adjacent to s. In particular C must contain at least four vertices.

We distinguish two cases. First, assume that C contains s. Then, there exists a vertex v in C that is not adjacent to s. Let u be a vertex in C that is adjacent to s (hence $u \neq v$), and let $S = V(C) \setminus \{s, u, v\}$. Then, it holds that

$$\begin{aligned}
\mathsf{mcst}(S \cup \{u\}) &= a(|S| + 1), \\
\mathsf{mcst}(S \cup \{v\}) &= a(|S| + 1), \\
\mathsf{mcst}(S \cup \{u, v\}) &= a(|S| + 2), \\
\mathsf{mcst}(S) &= a(|S| - 1) + b,
\end{aligned}$$

since C is a minimal cycle. Therefore

$$\mathsf{mcst}(S \cup \{u\}) + \mathsf{mcst}(S \cup \{v\}) - \mathsf{mcst}(S \cup \{u, v\}) - \mathsf{mcst}(S) = a - b < 0,$$

and thus (N, mcst) is not submodular.

For the second case, assume that C does not contain s. We may also assume that there is at most one vertex in C that is adjacent to s since otherwise the situation is reduced to the first case. Then, let u, v be a non-adjacent pair of vertices in C. Assume that v is not adjacent to s since at most one of u and v is adjacent to s. Then, denote by x, y the vertices adjacent to v in C. Note that x, y are not identical to u since the length of C is at least four. Let $S = V(C) \setminus \{x, y\}$. Then, it holds that

$$\begin{aligned}
\mathsf{mcst}(S \cup \{x\}) &= a|S| + \min_{z \in S \cup \{x\}} w(s, z), \\
\mathsf{mcst}(S \cup \{y\}) &= a|S| + \min_{z \in S \cup \{y\}} w(s, z), \\
\mathsf{mcst}(S \cup \{x, y\}) &= a(|S| + 1) + \min_{z \in S \cup \{x, y\}} w(s, z), \\
\mathsf{mcst}(S) &= a(|S| - 2) + b + \min_{z \in S} w(s, z)
\end{aligned}$$

since there is at most one vertex in C that is adjacent to s, as we have already assumed. Therefore,

$$\begin{aligned}
&\mathsf{mcst}(S \cup \{x\}) + \mathsf{mcst}(S \cup \{y\}) - \mathsf{mcst}(S \cup \{x, y\}) - \mathsf{mcst}(S) \\
&\quad = a - b + \min_{z \in S \cup \{x\}} w(s, z) + \min_{z \in S \cup \{y\}} w(s, z) - \min_{z \in S \cup \{x,y\}} w(s, z) - \min_{z \in S} w(s, z) \\
&\quad < 0,
\end{aligned}$$

since we can observe that

$$\min_{z \in S \cup \{x\}} w(s, z) + \min_{z \in S \cup \{y\}} w(s, z) - \min_{z \in S \cup \{x,y\}} w(s, z) - \min_{z \in S} w(s, z) = 0$$

by case analysis (distinguishing which vertex in C is adjacent to s in G'). This proves the only-if part. $\square$

5.2 Proof of the if part

Proof of the if part of Theorem 1.3. We now turn to the if part. Assume that the vertices of every cycle C in G' are adjacent to s or they are pairwise adjacent. As in the proof of Theorem 1.1, we verify the following condition equivalent to submodularity: for every $i, j \in N$ and $S \subseteq N \setminus \{i, j\}$,

$$\mathsf{mcst}(S \cup \{i\}) + \mathsf{mcst}(S \cup \{j\}) \geq \mathsf{mcst}(S) + \mathsf{mcst}(S \cup \{i, j\}).$$

Therefore, choose $i, j \in N$ and $S \subseteq N \setminus \{i, j\}$ arbitrarily.

Consider a minimum-cost spanning tree T of $G[S \cup \{s, i\}]$ that has the fewest edges incident to i. By the choice of T, we see that the set of edges incident to i in T consists of either one edge of weight b, or several edges of weight a. Namely, if i is incident to at least two edges of weight b in T, then removing one of these two edges from T and adding another edge not incident to i yields another minimum-cost spanning tree with fewer edges incident to i. We distinguish two cases.

First assume that there is only one edge e incident to i in T (no matter which weight e has, a or b). Then, $T - e$ is a minimum-cost spanning tree of $G[S \cup \{s\}]$. Therefore, $\mathsf{mcst}(S) = \mathsf{mcst}(S \cup \{i\}) - w(e)$. On the other hand, let T' be a minimum-cost spanning tree of $G[S \cup \{s, j\}]$. Then, $T' + e$ is a spanning tree of $G[S \cup \{s, i, j\}]$. Therefore, $\mathsf{mcst}(S \cup \{i, j\}) \leq \mathsf{mcst}(S \cup \{j\}) + w(e)$. Consequently, we obtain

$$\begin{aligned}
\mathsf{mcst}(S) + \mathsf{mcst}(S \cup \{i, j\}) &\leq \mathsf{mcst}(S \cup \{i\}) - w(e) + \mathsf{mcst}(S \cup \{j\}) + w(e) \\
&= \mathsf{mcst}(S \cup \{i\}) + \mathsf{mcst}(S \cup \{j\}),
\end{aligned}$$

and so the submodular inequality is satisfied.

For the second case, we assume that there are at least two edges incident to i in T. Then, according to the observation above, these edges must have weight a. Let k be the number of edges incident to i in T. If we remove i from T, then the result $T - i$ consists of k connected components each of which is a tree. We can observe that there is no edge of weight a between any two such connected components due to the choice of T (such that the number of edges incident to i is minimized). Therefore, in any minimum-cost spanning tree of $G[S \cup \{s\}]$ we must use $k - 1$ edges of weight b between these connected components. Thus we have

$$\mathsf{mcst}(S) = \mathsf{mcst}(S \cup \{i\}) - ak + b(k - 1).$$

Now let T' be a minimum-cost spanning tree of $G[S \cup \{s, j\}]$. We consider adding the edges incident to i in T to T' one by one to create a spanning tree of $G[S \cup \{s, i, j\}]$ of small cost. Adding one edge, we obtain a spanning tree of $G[S \cup \{s, i, j\}]$. Adding more edges, we create a cycle C and so we remove one edge every time to keep the graph a tree. Note that the created cycle C always traverse i.

We claim that the cycle C always contains an edge of weight b. To this end, suppose that the weight of every edge of C is a. Then, C is a cycle of G' (as in the statement of the theorem). By the assumption, all vertices of C are adjacent to s in G' , or all vertices of C are pairwise adjacent in G'. However, this is impossible by the following reason. Look at how C traverses several connected components of $T - i$. We start at i, and then enters a connected component of $T - i$. To get back to i, C needs to go through another connected component of $T - i$, and arrives at i through a (unique) edge from that component to i. If C does not go through j, then C must use an edge between two connected components of $T - i$. However, such an edge costs b as we have already observed. Therefore, C must go through j. Namely, C is a cycle of length at least four, and the weight of every edge of C is a. The cycle C contains at least two vertices x, y from different connected components of $T - i$, which means that $w(x, y) = b$ and $\{x, y\}$ is not an edge of G'. Namely, the vertices of C are not pairwise adjacent in G', and hence by the assumption, the vertices of C are adjacent to s in G'. Since at least one of x and y (say x) belongs to a connected component of $T - i$ that does not contain s, by the observation above it holds that $w(s, x) = b$. However, since s and x are adjacent in G', we have $w(s, x) = a$. This is a contradiction. This proves the claim.

Hence, when we remove an edge from C, we can always choose an edge of weight b. Therefore, after adding k edges incident to i in T to T', we remove $k - 1$ edges of weight b. This implies that

$$\mathsf{mcst}(S \cup \{i, j\}) \leq \mathsf{mcst}(S \cup \{j\}) + ak - b(k - 1).$$

Consequently, we obtain

$$\begin{aligned}\mathsf{mcst}(S) + \mathsf{mcst}(S \cup \{i, j\}) &\leq \mathsf{mcst}(S \cup \{i\}) - ak + b(k - 1) + \mathsf{mcst}(S \cup \{j\}) + ak - b(k - 1)\\ &= \mathsf{mcst}(S \cup \{i\}) + \mathsf{mcst}(S \cup \{j\}),\end{aligned}$$

and so the submodular inequality is satisfied. This completes the proof of the if of the theorem. □

The algorithmic part is not difficult. We only need to construct a biconnected-component decomposition, which can be found in $O(n^2)$ time [15], then examine whether each biconnected component satisfies the condition. This is enough since every cycle is contained in a biconnected component.

6 Concluding remarks

The conditions in Theorems 1.1 and 1.2 are similar. However, we have not succeeded in giving a necessary and sufficient condition for submodularity. This is the main open question. As we have already mentioned in Section 1, we feel that recognizing a submodular minimum-cost spanning tree game is coNP-complete, but we are still far from proving such a result.

Having Theorem 1.3, one may wonder what if the weights are restricted to three values. This would be completely different from what we saw in Theorem 1.3 since in that theorem we only need to look at the graph G', but if we have three values the graph structure of G' (or similar) would not be enough.

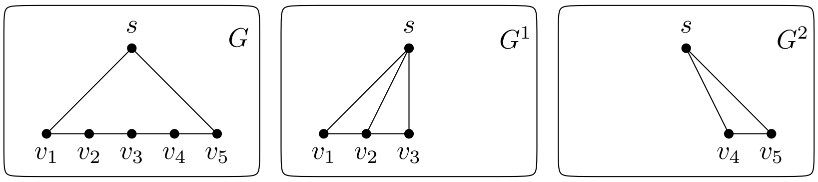

Figure 2: An example for which a decomposition result fails. The edges show pairs with weight 1, and pairs not connected by edges have weight 5.

Granot and Huberman [5] proved a decomposition theorem of the core and the nucleolus of minimum-cost spanning tree games. If a similar decomposition theorem would exist for submodularity, then the investigation could be easier. However, we give an example for which such a result does not hold. To this end, we need to define a decomposition more formally.

Assume that a minimum-cost spanning tree T of G has two edges e_1 and e_2 incident to s. Let $N_1, N_2 \subseteq N$ be the set of vertices of G that have paths to 0 through e_1, e_2 in T, respectively. Note that N_1 and N_2 partitions N. Then, we consider two graphs $G^1 = G[N_1 \cup \{s\}]$ and $G^2 = G[N_2 \cup \{s\}]$, and two functions w^1 on the edge set of G^1 and w^2 on the edge set of G^2 as follows:

$$w^1(\{u,v\}) = \begin{cases} \min\{w(\{u,v\}), \min\{w(\{s,v'\}) \mid v' \in N_2\}\} & \text{if } u = s \text{ or } v = s, \\ w(\{u,v\}) & \text{otherwise;} \end{cases}$$

$$w^2(\{u,v\}) = \begin{cases} \min\{w(\{u,v\}), \min\{w(\{s,v'\}) \mid v' \in N_1\}\} & \text{if } u = s \text{ or } v = s, \\ w(\{u,v\}) & \text{otherwise.} \end{cases}$$

Let (N_1, mcst^1) be the minimum-cost spanning tree game arising from G^1 and w^1, and (N_2, mcst^2) be the minimum-cost spanning tree game arising from G^2 and w^2. Granot and Huberman [5] proved that the core of (N, mcst) is the Cartesian product of the cores of (N_1, mcst^1) and (N_2, mcst^2), and the nucleolus of (N, mcst) is the Cartesian product of the nucleoli of (N_1, mcst^1) and (N_2, mcst^2).

If a similar result would exist for submodularity, it would be read as "(N, mcst) is submodular if (N_1, mcst^1) and (N_2, mcst^2) are submodular." Below we give an example in which this statement fails to hold even when the weights are bound to two values.

Consider the following five-player situation. The vertex set of G is $\{s, v_1, v_2, v_3, v_4, v_5\}$, and the weight w is defined as:

$$w(\{s, v_i\}) = \begin{cases} 1 & \text{if } i \in \{1,5\}, \\ 5 & \text{if } i \in \{2,3,4\}, \end{cases}$$

$$w(\{v_i, v_j\}) = \begin{cases} 1 & \text{if } (i,j) \in \{(1,2),(2,3),(3,4),(4,5)\}, \\ 5 & \text{otherwise.} \end{cases}$$

Figure 2 is an illustration.

Then, the minimum-cost spanning tree game (N, mcst) is not submodular. Actually, for $S = \{v_1, v_2, v_3\}$ and $T = \{v_3, v_4, v_5\}$, we obtain $\mathsf{mcst}(S) = 3, \mathsf{mcst}(T) = 3, \mathsf{mcst}(S \cup T) = 5, \mathsf{mcst}(S \cap T) = 5$. On the other hand, we can see that for $N_1 = \{v_1, v_2, v_3\}$ and $N_2 = \{v_4, v_5\}$, the minimum-cost spanning tree games (N_1, mcst^1) and (N_2, mcst^2) are submodular.

Acknowledgments The authors are grateful to Toshihiro Fujito and the members of Discrete Optimization Laboratory, Department of Information and Computer Sciences, Toyohashi

University of Technology for constructive suggestions. They also thank the anonymous referees for helpful comments.

References

[1] C. Bird: On cost allocation for a spanning tree: a game theoretic approach. Networks **6** (1976) 335–350.

[2] A. Claus and D.J. Kleitman: Cost allocation for a spanning tree. Networks **3** (1973) 289–304.

[3] I.J. Curiel: Cooperative Game Theory and Applications: Cooperative Games Arising from Combinatorial Optimization Problems. Kluwer Academic Publishers, Dordrecht, 1997.

[4] X. Deng, T. Ibaraki, and H. Nagamochi: Algorithmic aspects of combinatorial optimization games. Mathematics of Operations Research **24** (1999) 751–766.

[5] D. Granot and G. Huberman: On minimum cost spanning tree games. Mathematical Programming **21** (1981) 1–18.

[6] D. Granot and G. Huberman: The relationship between convex games and minimum cost spanning tree games: A case for permutationally convex games. SIAM Journal of Algebraic and Discrete Methods **3** (1982) 288–292.

[7] J.B. Kruskal: On the shortest spanning subtree of a graph and the traveling salesman problem. Proceedings of the American Mathematical Society, **7** (1956) 48–50.

[8] J. Kuipers: A polynomial time algorithm for computing the nucleolus of convex games. Report M 96-12, Maastricht University, 1996.

[9] J. Kuipers: Minimum cost forest games. International Journal of Game Theory **26** (1997) 367–377.

[10] M. Maschler, B. Peleg, and L. Shapley: The kernel and bargaining set of convex games. International Journal of Game Theory **2** (1972) 73–93.

[11] N. Megiddo: Cost allocation for Steiner trees. Networks **8** (1978) 1–6.

[12] Y. Okamoto: Submodularity of some classes of the combinatorial optimization games. Mathematical Methods of Operations Research **58** (2003) 131–139.

[13] E.C. Rosenthal: The minimum cost spanning forest game. Economic Letters **23** (1987) 355–357.

[14] L. Shapley: Cores of convex games. International Journal of Game Theory **1** (1971) 11–26. Errata is in the same volume, 1972, p. 199.

[15] R.E. Tarjan: Depth-first search and linear graph algorithms. SIAM Journal on Computing **2** (1972) 146–160.

[16] S. Tijs: Bounds for the core and the τ-value. O. Moeschlin and P. Pallaschke, eds., Game Theory and Mathematical Economics, North Holland, Amsterdam, 1981, pp. 123–132.

[17] C. Trudeau: A new stable and more responsive cost sharing solution for minimum cost spanning tree problems. Games and Economic Behavior **75** (2012) 402–412.