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: Cooperative game; Minimum-cost spanning tree game; Network

1 Introduction

Some cost allocation problems can be modeled within the framework of cooperative game theory. Especially, 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.

Let us model the problem more formally. 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 $E = \binom{N \cup \{s\}}{2}$. 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. 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 cycle) 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 as small as possible 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. A minimum-cost spanning tree game was first introduced by Bird [1].

The aim of this work is to try to characterize minimum-cost spanning tree games with nice properties. As such a nice property, we concentrate on the submodularity. A minimum-cost spanning tree game (N, mcst) is *submodular* (or *convex*) if for every $S, T \subseteq N$ it holds

[*]Discrete Optimization Laboratory, Department of Information and Computer Sciences, Toyohashi University of Technology, Hibarigaoka 1-1, Tempaku, Toyohashi, Aichi, 441-8580, Japan

[†]Corresponding author. Graduate School of Information Science and Engineering, Tokyo Institute of Technology, 2-12-1-W8-88, Ookayama, Meguro-ku, Tokyo, 152-8552, Japan. Supported by Global COE Program "Computationism as a Foundation for the Sciences" and Grant-in-Aid for Scientific Research from Ministry of Education, Science and Culture, Japan, and Japan Society for the Promotion of Science.

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: A core is always non-empty and it is a unique von Neumann-Morgenstern solution [10]; The Shapley value is the barycenter of the vertices of the core (when the degeneracy is taken into account) [10]; The core and the bargaining set coincide and the kernel and the nucleolus coincide [8]; The nucleolus and the τ-value can be computed in polynomial time ([7] and [12] 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 unlikely to exist), but we are still far from conjecturing so.

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 for every minimum-spanning tree T of G, 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 and 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 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 vertex $u \in N$ on the (unique) path connecting s and v in T and for every 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 weight is bound to two values $w\colon E \to \{a,b\}$ where $a < b$. For this case, we are able to give a characterization for submodularity.

Theorem 1.3. *When the edge weight is restricted to two values $w\colon E \to \{a,b\}$ 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 & 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 & Huberman [5]. Since then, various aspects of minimum-cost spanning tree games have been investigated (we omit the vast literature). Among them, Granot & 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 game. 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 partial knowledge has been provided by the second author [9] who characterized submodular minimum coloring games and submodular minimum vertex cover games that were defined by Deng, Ibaraki & Nagamochi [4].

2 Proof of Theorem 1.1

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

Lemma 2.1. *Let T be an arbitrary minimum-cost spanning tree of $G = (N \cup \{s\}, E)$. If $w(s,v) \geq w(s,u)$ for every vertex $v \in N$ and every vertex $u \in N$ on the (unique) path connecting s and v in T, and $w(s,v) \leq w(u,v)$ for every $\{u,v\} \notin T$, then a minimum-cost spanning tree T_S of $G[S \cup \{s\}]$ can be obtained in the following manner. For each $v \in S$, let $p(v) \in N \cup \{s\}$ be the vertex next to v on the path connecting s and v in T. Then we set $E(T_S) = \{\{v, p(v)\} \mid v \in S, p(v) \in S\} \cup \{\{s,v\} \mid v \in S, p(v) \notin S\}$.*

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. Let us go to the induction step. If $|E(T'_S) \setminus E(T_S)| > 0$, 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 $\{s, v\} \notin E(T'_S)$.

Suppose 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. Since T is a minimum-cost spanning tree of G and $\{v, p(v)\} \in E(T)$, There must be an edge e of C that does not take part in 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 $|E(T'_S + \{v, p(v)\} - e) \setminus E(T_S)| = |E(T'_S) \setminus E(T_S)| - 1$, by the induction hypothesis, the claim holds.

For the second case, suppose 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. The edges of $C - \{s, v\}$ are those on the unique path P from v to s in T'_S. Since $p(v) \notin S$, there must be an edge of P that does not take part in 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 are completely contained in T. Then, by the second assumption of the lemma and the observation that $x \neq s$, it holds that $w(x,y) \geq w(s,x)$. Now observe that v is on the unique path from x to s since $p(v)$ does not belong to P. Hence, by the first assumption 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 further $|E(T'_S + \{s,v\} - \{x,y\}) \setminus E(T_S)| = |E(T'_S) \setminus E(T_S)| - 1$. 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. 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 2.1 from T.

Denote by U the set of vertices u such that v lies on the unique path from u to s in T, and u is next to v 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 2.1. 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 a vertex 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 2.1 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 2.2. *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 vertex $u \in N$ on the path connecting s and 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 vertex $u \in N$ on the path connecting s and 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 2.2 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 2.2. The direction "1 ⇒ 2" is trivial. We prove the other direction. Assume that the statement 2 holds. Let T be a minimum-cost spanning tree of G in the 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. 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. Without loss of generality, we assume that u is on the path from v to s in T' and p is on the path from r to s in T. Let P be the path from v to s in T. Then, the edge $\{p, r\}$ belongs to P. By the first condition in the statement 2, it holds that $w(s, v) \geq w(s, r) \geq w(s, p)$. Furthermore, it holds that $w(u, v) \geq w(s, u)$ and $w(u, v) \geq w(s, v)$ by the second condition of the 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 condition for T, 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. □

3 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, there exist a vertex $v \in N$ and a vertex $u \in N$ on the unique path connecting s and v in T such that $w(s,v) < w(s,u)$, or 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 a vertex $u \in N$ on the unique path connecting s and 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 us choose v the furthest 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 which contains u and on which every vertex x satisfies $w(s,x) = w(s,u)$. If we denote the endpoint of P that is not u by r, we see that there exists a vertex on a path from r to s in T that is not r or s. Let p be a 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 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.

As 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 through s.

Let P be a maximal subpath of the path connecting v and s which contains v and on which every vertex x satisfies $w(s,x) = w(s,v)$. If we denote the endpoint of P that is not v by r, we see that there exists a vertex on a path from r to s in T that is not r or s. 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 path from u to s in T. Then, 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. □

4 Proof of Theorem 1.3

First notice that the submodularity does not depend on the values a, 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, so in the submodular inequality the total numbers of edges involved are identical on both sides.

Proof 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. Let u, v be a non-adjacent pair of vertices in C, 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.

As 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, again let u, v be a non-adjacent pair of vertices in C. Let us assume that u 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. 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,

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

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 (which vertex in C is adjacent to s in G'). This proves the only-if part.

Let us 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. 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

$$\mathsf{mcst}(S \cup \{i\}) + \mathsf{mcst}(S \cup \{j\}) - \mathsf{mcst}(S) - \mathsf{mcst}(S \cup \{i, j\}) \geq 0,$$

and so the submodular inequality is satisfied.

As 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 C goes through i.

We claim that such a cycle C always contains an edge of weight b. To this end, suppose that the weight of every edge of C is a. So C is a cycle of G' (as in the statement of the theorem). By the assumption all vertices of C are adjacent to s, or all vertices are pairwise adjacent. 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 any edge of C is a. The cycle C contains at least two vertices x, y from different connected components of $T - i$. If C contains s, then at least one of x, y (say x) belongs to a connected component that does not contain s. Then $w(s, x)$ must be b (by the observation above) but they must be adjacent in G' (by the assumption), which means that $w(s, x) = a$. This is a contradiction. If C does not contain s, then x, y must be adjacent in G' as well (by the assumption), and it leads to the contradiction in the same manner. 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

$$\mathsf{mcst}(S \cup \{i\}) + \mathsf{mcst}(S \cup \{j\}) - \mathsf{mcst}(S) - \mathsf{mcst}(S \cup \{i, j\}) \geq 0,$$

and so the submodular inequality is satisfied. This completes the proof of the first part of the theorem.

The second part (the algorithmic part) is not difficult. We only need to construct a biconnected-component decomposition, which can be found in $O(n^2)$ time [11], then examine whether each biconnected component satisfies the condition. This is enough since every cycle is embraced in a biconnected component. □

5 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 conjecturing so.

Having Theorem 1.3, one may wonder what if the weight is 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.

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.

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

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

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

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

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

[12] 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.