JORC (2013) 1:405–413
DOI 10.1007/s40305-013-0025-8

Computational Complexity of a Solution for Directed Graph Cooperative Games

Ayumi Igarashi · Yoshitsugu Yamamoto

Received: 24 January 2013 / Revised: 23 August 2013 / Accepted: 24 August 2013 /
Published online: 13 September 2013

Abstract Khmelnitskaya et al. have recently proposed the average covering tree value as a new solution concept for cooperative transferable utility games with directed graph structure. The average covering tree value is defined as the average of marginal contribution vectors corresponding to the specific set of rooted trees, and coincides with the Shapley value when the game has complete communication structure. In this paper, we discuss the computational complexity of the average covering tree value. We show that computation of the average covering tree value is $\#P$-complete even if the characteristic function of the game is $\{0,1\}$-valued. We prove this by a reduction from counting the number of all linear extensions of a partial order, which has been shown by Brightwell et al. to be a $\#P$-complete counting problem. The implication of this result is that an efficient algorithm to calculate the average covering tree value is unlikely to exist.

Keywords $\#P$-complete · Digraph game · Average covering tree value · Coalition · Communication structure · Linear extension

Mathematics Subject Classification (2010) 05C57 · 91A43 · 03D15 · 68Q17

This work was partially supported by the Okawa Foundation for Information and Telecommunication.

A. Igarashi (✉)
Graduate School of Systems and Information Engineering, University of Tsukuba, Tsukuba, Ibaraki 305-8573, Japan
e-mail: igarashi80@sk.tsukuba.ac.jp

Y. Yamamoto
Faculty of Engineering, Information and Systems, University of Tsukuba, Tsukuba, Ibaraki 305-8573, Japan
e-mail: yamamoto@sk.tsukuba.ac.jp

1 Introduction

The central question of cooperative game theory is how to distribute the total benefit gained from cooperative behavior of players, and many solution concepts have been proposed to date. Computational complexity is one of the most important criteria to judge whether a given solution concept is appropriate. A solution concept is not applicable in real settings if it requires computation time proportional to an exponential of the problem size; see, for example, [3].

This paper studies cooperative games with directed graph structure from the viewpoint of computational complexity. Khmelnitskaya et al. [5] introduced a cooperative game on a directed graph that furnishes the game with a communication structure. They proposed a single-valued solution concept, and named it the *average covering tree value*. To construct the average covering tree value, they introduced a so-called covering tree of a directed graph. The average covering tree value is defined as the average of marginal contribution vectors, each of which corresponds to a covering tree. The solution coincides with the Shapley value [7] when the game has a complete communication structure. We can apply the average covering tree value to games on undirected graphs by replacing every undirected link with two oppositely-directed links. The average covering tree value for this class of game is equal to the GC solution proposed by Koshevoy and Talman [6].

Concerning the computational complexity issue, we show in this paper that the problem of calculating the average covering tree value is $\#P$-complete. The proof uses a reduction from counting the number of all linear extensions of an arbitrary partial order, which has been shown by Brightwell et al. to be a $\#P$-complete counting problem. This result implies that an efficient algorithm to calculate the average covering tree value is unlikely to exist.

2 Preliminaries

2.1 TU-Games with Directed Graph Structure

We consider a cooperative transferable utility game with restricted communication structure, called *digraph games*. A digraph game is represented by a triple (N, v, Γ), where N is a finite set of n players, $v : 2^N \to \mathbb{R}$ is a *characteristic function*, and Γ is a subset of the set of all ordered pair of nodes, which represents directed communication links between players. A subset $S \in 2^N$ is called a *coalition* and $v(S)$ stands for the *worth* of a coalition S. A *payoff vector* $x \in \mathbb{R}^n$ is an n-dimensional vector giving payoff x_i to player $i \in N$.

2.2 Definitions for Digraph

The pair $G = (N, \Gamma)$ is called a *digraph* where N is a finite set of *nodes* and Γ is a collection of *directed links* between nodes. In this paper, we only consider digraphs without self-loops. For a digraph $G = (N, \Gamma)$, a sequence of different nodes $(i_1, i_2, \cdots, i_k)$, $k \geqslant 2$, is a *path* in Γ if $\{(i_h, i_{h+1}), (i_{h+1}, i_h)\} \cap \Gamma \neq \emptyset$

for $h = 1, 2, \cdots, k-1$. A path is said to be a *directed path* if $(i_h, i_{h+1}) \in \Gamma$ for $h = 1, 2, \cdots, k-1$. A path is said to be a *cycle* in Γ if $\{(i_k, i_1), (i_1, i_k)\} \cap \Gamma \neq \emptyset$, and a directed path is said to be a *directed cycle* in Γ if $(i_k, i_1) \in \Gamma$. A digraph $G = (N, \Gamma)$ is said to be *acyclic* if it has no directed cycles, and to be *transitive* if for all $i, j, k \in N$, $(i, j) \in \Gamma$ and $(j, k) \in \Gamma$ imply $(i, k) \in \Gamma$. For a digraph $G = (N, \Gamma)$, the subset of Γ induced by $S \in 2^N$ is defined as

$$\Gamma|S := \{(i, j) \in \Gamma \mid i, j \in S\}.$$

A subset $S \in 2^N$ is *connected* if for any two distinct nodes $i, j \in S$ there is a path in $\Gamma|S$ between i and j. For $S \in 2^N$, a subset K of S is called a *connected component* of S if K is maximally connected, i.e., K is connected but the set $K \cup \{j\}$ is not connected for any $j \in S \setminus K$. In this paper, we assume that N is always connected in the digraph (N, Γ).

For each node $i \in N$ of a digraph $G = (N, \Gamma)$ we define the set of *children* $\mathrm{ch}(\Gamma, i)$ and the set of *successors* $\mathrm{scc}(\Gamma, i)$ as

$$\mathrm{ch}(\Gamma, i) = \{j \in N \mid (i, j) \in \Gamma\}$$

and

$$\mathrm{scc}(\Gamma, i) = \{j \in N \mid \text{there exists a directed path from } i \text{ to } j \text{ in } \Gamma\}.$$

We also write

$$\overline{\mathrm{scc}}(\Gamma, i) = \{i\} \cup \mathrm{scc}(\Gamma, i).$$

A node $i \in N$ is said to be a *predecessor* of $j \in N$ in Γ if $j \in \mathrm{scc}(\Gamma, i)$. We denote the set of predecessors of j by $\mathrm{prd}(\Gamma, j)$, i.e.,

$$\mathrm{prd}(\Gamma, j) = \{i \in N \mid j \in \mathrm{scc}(\Gamma, i)\}.$$

An acyclic connected digraph (N, T) is said to be a *tree* if it has a unique node without predecessors, the *root*, and for every other node in N there is a unique directed path in T from the root to that node. A node $i \in S$ is an *undominated* node of S if $\mathrm{prd}(\Gamma|S, i) \subseteq \mathrm{scc}(\Gamma|S, i)$, and the set of undominated nodes of S is denoted by $U(\Gamma, S)$, i.e.,

$$U(\Gamma, S) = \{i \in S \mid \mathrm{prd}(\Gamma|S, i) \subseteq \mathrm{scc}(\Gamma|S, i)\}.$$

A node $i \in S$ is a *nondominant* node of S if $\mathrm{scc}(\Gamma|S, i) \subseteq \mathrm{prd}(\Gamma|S, i)$, and the set of non dominant nodes of S is denoted by $D(\Gamma, S)$, i.e.,

$$D(\Gamma, S) = \{i \in S \mid \mathrm{scc}(\Gamma|S, i) \subseteq \mathrm{prd}(\Gamma|S, i)\}.$$

A node $i \in N$ is called a *minimum* node of (N, Γ) if $i \in \mathrm{scc}(\Gamma, j)$ for every $j \in N \setminus \{i\}$. If an acyclic digraph has a minimum node, it is uniquely determined.

2.3 Definitions for Poset

A *partially ordered set*, or *poset* for short, is a pair (N, Γ) where N is a finite set and Γ is a *partial order* on N, that is, an irreflexive, antisymmetric, and transitive binary relation. Two elements i and j are *comparable* in Γ if either $(i, j) \in \Gamma$ or $(j, i) \in \Gamma$. A *linear order* on N is a partial order Γ in which every pair of elements of N is comparable. A *linear extension* of a partial order Γ on N is a linear order $\overline{\Gamma}$ on N such that $(i, j) \in \overline{\Gamma}$ whenever $(i, j) \in \Gamma$. Equivalently, a linear extension of a partial order Γ on N is a bijection π from N to $\{1, 2, \cdots, |N|\}$ such that for all $i, j \in N$, $(i, j) \in \Gamma$ implies $\pi(i) < \pi(j)$.

2.4 Digraphs and Posets

Every poset (N, Γ) corresponds to a digraph by considering N as the set of nodes and Γ as the set of directed links. This digraph is acyclic and transitive. Conversely, for every acyclic transitive digraph $G = (N, \Gamma)$, Γ is a partial order on N.

Lemma 2.1 (Bondy [1]) *A digraph G is a poset if and only if G is acyclic and transitive.*

3 The Average Covering Tree Value

3.1 Covering Trees

In this section, we provide the definition of the average covering tree value introduced by Khmelnitskaya et al. [5]. The average covering tree value is the average of marginal contribution vectors with respect to specific trees, called *covering trees* of $G = (N, \Gamma)$. They propose to construct a covering tree of G by Algorithm 3.1 given below.

After initialization in Step 1, a node $i \in U(\Gamma, N)$ is taken as the root of a covering tree in Step 2. In the next iteration step after node i is deleted from N, the algorithm selects an undominated node of each connected component, and expands the tree T towards the selected undominated nodes from node i. Setting $Q(i)$ be empty, the algorithm proceeds to Step 4. As long as there remains a nonempty $Q(j)$, this procedure is repeated, and ends up having explored all the nodes.

It is clear that the output T is a rooted tree that spans N. More importantly, since this algorithm grows the tree by adding undominated nodes of remaining connected

Algorithm 3.1 Construction of a covering tree T of G

1: Set $T = \emptyset$ and $Q(j) = \emptyset$ for all $j \in N$.
2: Choose any $i \in U(\Gamma, N)$ and set $Q(i) = N \setminus \{i\}$.
3: Let $\{K_1, K_2, \cdots, K_m\}$ be the set of connected components of $Q(i)$. For every $k = 1, 2, \cdots, m$, choose $j_k \in U(\Gamma, K_k)$ and set $Q(j_k) = K_k \setminus \{j_k\}$. Set $T = T \cup \{(i, j_1), (i, j_2), \cdots, (i, j_m)\}$ and $Q(i) = \emptyset$.
4: If $Q(j) = \emptyset$ for all $j \in N$, then stop. Otherwise, choose $i \in N$ such that $Q(i) \neq \emptyset$ and return to Step 3.

Fig. 1 G_1: Acyclic transitive digraph with a minimum node and its covering trees

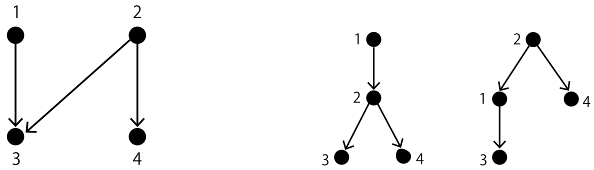

Fig. 2 G_2: Acyclic transitive digraph without a minimum node and its covering trees

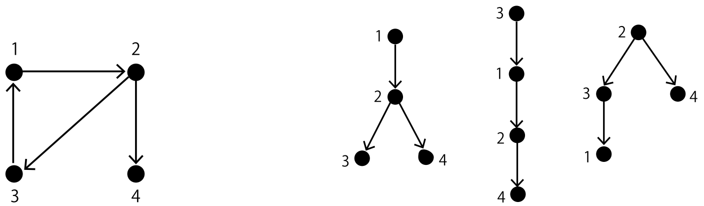

Fig. 3 G_3: Not acyclic digraph with a minimum node and its covering trees

components, the covering tree T preserves the subordination among nodes. This property is stated as the following lemma in [5].

Lemma 3.1 (Khmelnitskaya et al. [5]) *Let T be a covering tree of a digraph* (N, Γ). *If* $(i, j) \in \Gamma$ *and* $i \notin \mathrm{scc}(\Gamma, j)$, *then* $\overline{\mathrm{scc}}(\Gamma, j) \subseteq \mathrm{scc}(T, i)$.

Example 3.1 Three digraphs and their covering trees are given in Figs. 1, 2 and 3. The digraph G_1 is acyclic and transitive, and has a minimum node 3. Observe that every covering tree forms a directed path. G_2 is acyclic and transitive, but has no minimum node. The last digraph G_3 has a minimum node 4 but is neither acyclic nor transitive.

As seen in the above examples, covering trees may have different structures according to digraphs considered and the root chosen. We will discuss in more detail in the next section. We denote by $\mathcal{T}(\Gamma)$ the set of all covering trees of a digraph $G = (N, \Gamma)$ constructed by Algorithm 3.1.

Definition 3.1 (Marginal Contribution Vector) For a digraph game (N, v, Γ), the *marginal contribution vector* m^T corresponding to a covering tree $T \in \mathcal{T}(\Gamma)$ is the vector of payoffs given by

$$m_i^T = v\big(\overline{\mathrm{scc}}(T, i)\big) - \sum_{j \in \mathrm{ch}(T, i)} v\big(\overline{\mathrm{scc}}(T, j)\big) \tag{3.1}$$

for all $i \in N$.

Definition 3.2 (Average Covering Tree Value) For a digraph game (N, v, Γ), the *average covering tree value* is the average of the marginal contribution vectors m^T with respect to all covering trees of the digraph (N, Γ), i.e.,

$$ACT(N, v, \Gamma) = \frac{1}{|\mathcal{T}(\Gamma)|} \sum_{T \in \mathcal{T}(\Gamma)} m^T. \tag{3.2}$$

3.2 Properties of Covering Trees

We give some properties of covering trees provided that a digraph is acyclic.

Lemma 3.2 *Suppose $G = (N, \Gamma)$ is an acyclic digraph. Then, $i \in U(\Gamma, S)$ if and only if there is no node $j \in S$ such that $(j, i) \in \Gamma|S$.*

Proof (If): It holds from the definition of undominated node.

(Only-if): Let $i \in U(\Gamma, S)$. Assume that there exists a node $j \in S$ such that $(j, i) \in \Gamma|S$. Then $j \in \text{prd}(\Gamma|S, i)$, which implies $j \in \text{scc}(\Gamma|S, i)$ by the definition of undominated node. Thus a directed path from i to j in $\Gamma|S$ and the directed link (j, i) form a directed cycle in G, contradicting the fact that G is acyclic. □

Lemma 3.3 *Suppose $G = (N, \Gamma)$ is an acyclic digraph. Then, node $i \in D(\Gamma, S)$ if and only if there is no node $j \in S$ such that $(i, j) \in \Gamma|S$.*

Proof (If): It holds from the definition of nondominant node.

(Only-if): Let $i \in D(\Gamma, S)$ and assume that there exists a node $j \in S$ such that $(i, j) \in \Gamma|S$. Then $j \in \text{scc}(\Gamma|S, i)$, hence we see $j \in \text{prd}(\Gamma|S, i)$ since $i \in D(\Gamma, S)$. Thus a directed path from j to i in $\Gamma|S$ and the directed link (i, j) form a directed cycle in G, contradicting the fact that G is acyclic. □

Lemma 3.4 *Suppose $G = (N, \Gamma)$ is an acyclic and transitive digraph with a minimum node. Then the covering tree provided by Algorithm 3.1 is a directed path, which yields a linear extension of Γ.*

Proof First, note that G is a poset by Lemma 2.1. Let i^* be a minimum node of G. Node i^* will not be selected as j_k at Step 3 of Algorithm 3.1 unless all other nodes have been chosen. Thus $Q(i)$ is kept connected via i^*, and Algorithm 3.1 grows the tree by adding a single node at every iteration. Therefore, the final output T forms a directed path. Denote T as $\{(i_k, i_{k+1}) \mid k = 1, 2, \cdots, n-1\}$, where $i_n = i^*$. To show that the sequence $(i_1, i_2, \cdots, i_n)$ is a linear extension of Γ, we assume $(i_k, i_l) \in \Gamma$ for some k and l such that $k > l$. Let $S = N \setminus \{i_1, \cdots, i_{l-1}\} = \{i_l, \cdots, i_n\}$. Then $i_k \in S$ and by construction $i_l \in U(\Gamma, S)$. This contradicts Lemma 3.2. □

Lemma 3.5 *Suppose $G = (N, \Gamma)$ is an acyclic and transitive digraph with a minimum node. Then, by making all possible choices of undominated nodes in Step 2 and Step 3, Algorithm 3.1 yields all linear extensions of Γ.*

Proof We have seen in Lemma 3.4 that Algorithm 3.1 yields a linear extension of Γ. We will show that any linear extension of Γ can be produced by Algorithm 3.1. Let $(i_1, i_2, \cdots, i_n)$ be any linear extension of Γ. Then $(i_k, i_1) \notin \Gamma$ for any $k > 1$, meaning $i_1 \in U(\Gamma, N)$. Hence Algorithm 3.1 can choose i_1 at Step 2. We suppose that Algorithm 3.1 has grown the tree T in the order of $i_1, i_2, \cdots, i_{h-1}$, and show that i_h can be one of the candidate nodes to be chosen at the next iteration. After i_{h-1} is selected as j_k, $Q(i_{h-1})$ is set to $N \setminus \{i_1, i_2, \cdots, i_{h-1}\} = \{i_h, i_{h+1}, \cdots, i_n\}$ and $Q(j) = \emptyset$ for all other node j. Hence Algorithm 3.1 chooses $Q(i_{h-1})$ as $Q(i)$ at Step 4 and returns to Step 3. Since $Q(i_{h-1})$ is connected through a minimum node of G, $Q(i_{h-1})$ itself forms a connected component of itself. Since $(i_1, i_2, \cdots, i_n)$ is a linear extension of Γ, $(j, i_h) \notin \Gamma$ for any node $j \in \{i_h, i_{h+1}, \cdots, i_n\}$. By Lemma 3.2, i_h is an undominated node of $\{i_h, i_{h+1}, \cdots, i_n\}$. Thus, Algorithm 3.1 can choose i_h and set $T = T \cup \{(i_{h-1}, i_h)\}$ at the next iteration. □

4 #P-Completeness of the Average Covering Tree Value

Given a partial order on the set of players, Faigle and Kern [4] generalized the Shapley value to games where a coalition is formed if and only if it is an ideal of the partial order. The generalized Shapley value is defined as the average of marginal contribution vectors over the set of all linear extensions of the given partial order. We have seen in Lemma 3.5 that the average covering tree value coincides with the generalized Shapley value when a digraph corresponds to a poset and contains a minimum node. Faigle and Kern considered games with $\{0, 1\}$-valued characteristic functions, and showed in Proposition 3 of [4] that computing the generalized Shapley value is $\#P$-complete. Their proof is based on a polynomial-time reduction from counting the number of linear extensions of a given partial order. We will prove the $\#P$-completeness of the average covering tree value in a similar way to their proof.

We start with giving another representation of the average covering tree value. For a poset (N, Γ), let $\mathcal{L}(\Gamma)$ denote the set of all linear extensions of Γ. We denote the cardinality of $\mathcal{L}(\Gamma)$ by $\ell(\Gamma)$ with the convention that $\ell(\emptyset) = 1$.

Lemma 4.1 *Let $G = (N, \Gamma)$ be an acyclic and transitive digraph with a minimum node, and (N, v, Γ) be a digraph game on G. Then the average covering tree value of player $i \in N$ is rewritten as*

$$ACT_i(N, v, \Gamma) = \frac{1}{\ell(\Gamma)} \sum_{\substack{S \subseteq N; \\ i \in U(\Gamma, S), \\ i \in D(\Gamma, (N \setminus S) \cup \{i\})}} \ell\big(\Gamma | (S \setminus \{i\})\big) \cdot \ell\big(\Gamma | (N \setminus S)\big) \cdot \big(v(S) - v\big(S \setminus \{i\}\big)\big). \quad (4.1)$$

Proof Take an arbitrary player i and fix it. Since G has a minimum node, Algorithm 3.1 produces all linear extensions of Γ. Hence, the average covering tree value of player i is given by

$$ACT_i(N, v, \Gamma) = \frac{1}{\ell(\Gamma)} \sum_{\pi \in \mathcal{L}(\Gamma)} \left(v\left(\{j \in N | \pi(i) \leqslant \pi(j)\}\right) - v\left(\{j \in N | \pi(i) \leqslant \pi(j)\} \setminus \{i\}\right)\right).$$

For each $S \subseteq N$ such that $i \in U(\Gamma, S)$ and $i \in D(\Gamma, (N \setminus S) \cup \{i\})$, there is the number

$$\ell\left(\Gamma \middle| (S \setminus \{i\})\right) \cdot \ell\left(\Gamma | (N \setminus S)\right)$$

of linear extensions $\pi \in \mathcal{L}(\Gamma)$ such that $\{j \in N \mid \pi(i) \leqslant \pi(j)\} = S$. Substituting this for (3.2) yields (4.1). □

Let (N, Γ) be an arbitrary poset. Adding a new element $i^* \notin N$ we extend it as

$$N^* = N \cup \{i^*\} \quad \text{and} \quad \Gamma^* = \Gamma \cup \{(j, i^*) \mid j \in N\},$$

and make a digraph $G^* = (N^*, \Gamma^*)$. Then G^* is acyclic and transitive, and has a minimum node. Let $(i_1, i_2, \cdots, i_{n+1})$ be an arbitrary linear extension of Γ^*, where $i_{n+1} = i^*$. Then let

$$N_k^* = \{i_k, i_{k+1}, \cdots, i_{n+1}\},$$
$$\Gamma_k^* = \Gamma^* | N_k^*,$$
$$\delta_k^*(S) = \begin{cases} 1 & \text{if } S = N_k^*, \\ 0 & \text{otherwise} \end{cases}$$

for $k = 1, 2, \cdots, n + 1$. Note that N_k^* is connected, (N_k^*, Γ_k^*) is a poset with a minimum node, and the leading node i_k is an undominated node of N_k^*, i.e., $i_k \in U(\Gamma_k^*, N_k^*)$.

Proposition 4.1 (#*P*-Completeness of the Average Covering Tree Value) Assume that for $k = 1, 2, \cdots, n$ there exists a polynomial-time algorithm to compute the average covering tree value for digraph games $(N_k^*, \delta_k^*, \Gamma_k^*)$ constructed from a given poset (N, Γ). Then there exists a polynomial-time algorithm to compute the number $\ell(\Gamma)$ of linear extensions of Γ.

Proof Since each N_k^* contains a minimum node i_{n+1}, the formula (4.1) yields

$$\begin{aligned} ACT_{i_k}&\left(N_k^*, \delta_k^*, \Gamma_k^*\right) \\ &= \frac{1}{\ell(\Gamma_k^*)} \sum_{\substack{S \subseteq N_k^*; \\ i_k \in U(\Gamma_k^*, S), \\ i_k \in D(\Gamma_k^*, (N_k^* \setminus S) \cup \{i_k\})}} \ell\left(\Gamma^* \middle| (S \setminus \{i_k\})\right) \cdot \ell\left(\Gamma^* \middle| (N_k^* \setminus S)\right) \\ &\quad \cdot \left(\delta_k^*(S) - \delta_k^*\left(S \setminus \{i_k\}\right)\right) \\ &= \frac{1}{\ell(\Gamma_k^*)} \ell\left(\Gamma^* \middle| (N_k^* \setminus \{i_k\})\right) \cdot \ell(\emptyset) = \frac{\ell(\Gamma_{k+1}^*)}{\ell(\Gamma_k^*)} \end{aligned} \tag{4.2}$$

for $k = 1, 2, \cdots, n + 1$. It follows that

$$ACT_{i_1} ACT_{i_2} \cdots ACT_{i_n} = \ell(\Gamma^*)^{-1}.$$

Since i^* is a minimum node of (N^*, Γ^*), we have

$$\ell(\Gamma^*) = \ell(\Gamma^* | N) = \ell(\Gamma).$$

Consequently, we obtain

$$ACT_{i_1} ACT_{i_2} \cdots ACT_{i_n} = \ell(\Gamma)^{-1}.$$

This proves the proposition. □

Proposition 4.1 states that we are able to compute $\ell(\Gamma)$ in polynomial-time if there exists a polynomial-time algorithm to compute $ACT_{i_k}(N_k^*, \delta_k^*, \Gamma|_{N_k^*})$ for $k = 1, 2, \cdots, n$. Brightwell and Winkler [2] proved that the problem of counting the number of linear extensions of an arbitrary partial order is $\#P$-complete. Therefore, computing the average covering tree value for this class of games is also $\#P$-complete, implying that an efficient algorithm is unlikely to exist.

Note that our main result is not a direct consequence of Proposition 3 in [4]. This is because the average covering tree value coincides with the generalized Shapley value only when the digraph has a minimum node. Adding a minimum node i^* to a given poset plays a crucial role in obtaining (4.2) in the proof of Proposition 4.1.

5 Conclusion

In this paper, we showed that computing the average covering tree value is $\#P$-complete even if the characteristic function of the game is $\{0, 1\}$-valued. Characterization of games whose average covering tree value can be calculated efficiently might be an interesting future research theme. It seems reasonable to conjecture that the average covering tree value can be computationally more tractable when the digraph has tree-like structure. Development of an efficient algorithm is also desirable.

Acknowledgements We wish to thank the two anonymous reviewers for their constructive suggestions and comments. The comments have helped us significantly improve the paper.

References

[1] Bondy, J.A., Murty, U.S.R.: Graph Theory. Springer, Berlin (2008)
[2] Brightwell, G., Winkler, P.: Counting linear extensions is $\#P$-complete. Order **8**(3), 175–181 (1991)
[3] Deng, X., Papadimitriou, Ch.H.: On the complexity of cooperative solution concepts. Math. Oper. Res. **19**, 257–266 (1994)
[4] Faigle, U., Kern, W.: The Shapley value for cooperative games under precedence constraints. Int. J. Game Theory **21**, 249–266 (1992)
[5] Khmelnitskaya, A.B., Selcuk, Ö., Talman, A.J.J.: The average covering tree value for directed graph games. CentER Discussion Paper 2012-037, CentER, Tilburg University, pp. 203–212 (2011)
[6] Koshevoy, G.A., Talman, A.J.J.: Solution concepts for games with general coalitional structure. CentER Discussion Paper 2011-119, CentER, Tilburg University (2011)
[7] Shapley, L.S.: A value for n-person games. In: Kuhn, H.W., Tucker, A.W. (eds.) Contributions to the Theory of Games II, pp. 307–317. Princeton University Press, Princeton (1953)