Department of Social Systems and Management

Discussion Paper Series

No.1302

Computational Complexity of the Average Covering Tree Value

by

Ayumi IGARASHI and Yoshitsugu YAMAMOTO

November 2012

UNIVERSITY OF TSUKUBA
Tsukuba, Ibaraki 305-8573
JAPAN

COMPUTATIONAL COMPLEXITY OF THE AVERAGE COVERING TREE VALUE

AYUMI IGARASHI AND YOSHITSUGU YAMAMOTO

Abstract. In this paper we prove that calculating the average covering tree value recently proposed as a single-valued solution of graph games is #P-complete.

1. Introduction

Khmelnitskaya et al.[4] introduced cooperative games with directed graph structure and proposed its single-valued solution concept, called the *average covering tree value*. The purpose of this paper is to demonstrate that a problem for calculating the average covering tree value is #P-complete.

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 $\Gamma \subseteq \{\, (i, j) \mid i \neq j, i, j \in N \,\}$ is a collection of 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* $\boldsymbol{x} \in \mathbb{R}^n$ 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. For a digraph $G = (N, \Gamma)$, a sequence of different nodes $(i_1, i_2, \ldots, i_k)$, $k \geq 2$, is a *path* in Γ if $\{(i_h, i_{h+1}), (i_{h+1}, i_h)\} \cap \Gamma \neq \emptyset$ for $h = 1, 2, \ldots, k-1$. A sequence $(i_1, i_2, \ldots, i_k)$, $k \geq 2$, is a *directed path* if $(i_h, i_{h+1}) \in \Gamma$ for all $h \in \{1, 2, \ldots, k-1\}$. A path $(i_1, i_2, \ldots, i_k)$ in Γ is a *cycle* in Γ if $\{(i_k, i_1), (i_1, i_k)\} \cap \Gamma \neq \emptyset$, and a directed path $(i_1, i_2, \ldots, i_k)$, $k \geq 2$, in Γ is 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. A digraph $G = (N, \Gamma)$ is said to be *transitive* if for all $i, j, k \in N$, $(i, j) \in \Gamma$ and $(j, k) \in \Gamma$ implies $(i, k) \in \Gamma$. The *transitive closure* of a digraph $G = (N, \Gamma)$ is the digraph $G^+ = (N, \Gamma^+)$ where

$$\Gamma^+ = \{\, (i, j) \mid \text{there is a directed path from } i \text{ to } j \text{ in } \Gamma \,\}.$$

It is clear that the digraph G^+ is transitive. 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*

Date: November 13, 2012.

This research is supported by JSPS KAKENHI Grant Number 22510136.

of S if K is maximally connected, i.e., K is connected but the set $K \cup \{j\}$ is not connected for any $j \in S \setminus K$. For a digraph $G = (N, \Gamma)$, for each node $i \in N$ we define its sets of *successors* and *descendants* as

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

and

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

A node $i \in N$ is said to be a *predecessor* of $j \in N$ in Γ if there exists a directed path from i to j in Γ. 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 for every predecessor j of i in $\Gamma|_S$ there exists a directed path in $\Gamma|_S$ from i to j. A node $i \in S$ is a *nondominant* node of S if for every descendants $j (\neq i)$ of i in $\Gamma|_S$, there exists a directed path in $\Gamma|_S$ from j to i. For a digraph (N, Γ) and a subset $S \in 2^N$, let $U_\Gamma(S)$ denote the set of undominated nodes of S and $D_\Gamma(S)$ denote the set of nondominant nodes of S. A node $i \in N$ is called the *minimum* node of (N, Γ) if for all $j \in N \setminus \{i\}$ there exists a directed path from j to i in (N, Γ). If an acyclic digraph has the minimum node, it is uniquely determined.

2.3. **Definitions for Poset.** A *partially ordered set*, or for short *poset* is a pair $P = (N, \Gamma)$, 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* if either $(i, j) \in \Gamma$ or $(j, i) \in \Gamma$. A *linear ordering* on a poset $P = (N, \Gamma)$ is a bijection π from N to $\{1, 2, \ldots, |N|\}$ such that for all $i, j \in N$, $(i, j) \in \Gamma$ implies $\pi(i) < \pi(j)$. For a poset $P = (N, \Gamma)$, let $\mathcal{R}(\Gamma)$ denote the set of all linear orderings, where $\mathcal{R}(\emptyset) = 1$.

2.4. **Digraphs and Posets.** Every poset $P = (N, \Gamma)$ corresponds to a digraph 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. *A digraph G is a poset, if and only if G is acyclic and transitive.*

In this paper it is assumed that without loss of generality N is always connected in the graph (N, Γ).

3. The average covering tree value

In this section we provide the definition of the average covering tree value, introduced by Khmelnitskaya et al.[4]. The average covering tree value is the average of marginal contribution vectors with respect to specific trees, called *covering trees* $G = (N, \Gamma)$. In order to construct a covering tree of G, Khmelnitskaya et al.[4] apply Algorithm 1 on the next page. We denote by $\mathcal{T}^{\Gamma}$ the set of all covering trees of a digraph G constructed by Algorithm 1.

Algorithm 1 Construct a covering tree of $G = (N, \Gamma)$

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 $\{K_1, K_2, \ldots, K_m\}$ be the set of connected components of Q_i. For every $k = 1, 2, \ldots, m$, choose $j_k \in U_\Gamma(K_k)$ and set $Q_{j_k} = K_k \setminus \{j_k\}$. Set $T = T \cup \{(i, j_1), (i, j_2), \ldots, (i, j_m)\}$ 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.

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

$$m_i^T = v(des^T(i)) - \sum_{j \in suc^T(i)} v(des^T(j)), \quad \text{for all } i \in N. \tag{3.1}$$

Definition 3.2 (ACT(N, v, Γ))**.** For a digraph game (N, v, Γ), the *average covering tree value* is the average of the marginal contribution vectors $\boldsymbol{m}^T$ with respect to all covering trees of the digraph Γ, i.e.,

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

4. Properties of a covering tree

In this section we provide some properties of the covering tree when the digraph is acyclic.

Lemma 4.1. *Given an acyclic digraph $G = (N, \Gamma)$. Node i is 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 an 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 there exists a directed path from i to j in $\Gamma|_S$, and (j, i) completes a directed cycle in G, contradicting the fact that G is acyclic. □

Lemma 4.2. *Given an acyclic digraph $G = (N, \Gamma)$. Node i is 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 a nondominant node.
(Only-if): Let $i \in D_\Gamma(S)$. Assume that there exists a node $j \in S$ such that $(i, j) \in \Gamma|_S$. Then there exists a directed path from j to i in $\Gamma|_S$, and (i, j) completes a directed cycle in G, contradicting the fact that G is acyclic. □

Lemma 4.3. *Given an acyclic transitive digraph $G = (N, \Gamma)$. Algorithm* 1 *yields a linear ordering on G if and only if G has the minimum node.*

Proof. First, note that G is a poset by Lemma 2.1.
(If): Suppose G has the minimum node i^*. Node i^* will not be selected as j_k at Step 3 of Algorithm 1 unless all the other nodes have been chosen by Algorithm 1. Thus for every iteration, any two nodes of Q_i are connected via i^*. Since Q_i is connected for each iteration, Algorithm 1 grows the tree T by adding only one node. Thus, the final output T of Algorithm 1 is denoted by a sequence $(i_1, i_2, \ldots, i_n)$, where $(i_k, i_{k+1}) \in T$ for all $k = 1, 2, \ldots, n-1$ and $i_n = i^*$. Next we will show that the sequence $(i_1, i_2, \ldots, i_n)$ is a linear ordering on G. Let $(i_k, i_{k'}) \in \Gamma$. Assume that $k > k'$. Node $i_{k'}$ is an undominated node of $N \setminus \{i_1, i_2, \ldots, i_{k'-1}\}$. Then from the assumption that $k > k'$, $i_k \in N \setminus \{i_1, i_2, \ldots, i_{k'-1}\}$. This leads to a contradiction to Lemma 4.1.
(Only-if): We will prove this part by contrapositive. Suppose G does not have the minimum node. Then there exist two different nondominant nodes i^*, j^* of N, i.e., $i^*, j^* \in D^\Gamma(N)$. Let K_k denote the connected component containing i^* when Algorithm 1 chooses i^* as j_k at Step 3, i.e., $i^* \in U_\Gamma(K_k)$. If K_k has another node $i'(\neq i^*)$, there is a path from i^* to i' in $\Gamma|_{K_k}$ since K_k is connected. By Lemma 4.1 $i^* \in U_\Gamma(K_k)$ implies that there is no node $j \in K_k$ such that $(j, i) \in \Gamma|_{K_k}$. Thus there is a node j' on the path between i^* and i' such that $(i^*, j') \in \Gamma|_{K_k}$, contradicting Lemma 4.2. Thus K_k only contains a node i^*. After i^* is chosen by Algorithm 1, Q_{i*} becomes empty. Hence there is no directed path from i^* to j^* in any covering tree T of (N, Γ). Similarly, there is no directed path from j^* to i^* in T. Nodes i^* and j^* are not comparable in (N, T^+). Algorithm 1 does not yield a linear ordering on G. □

Lemma 4.4. *Let $G = (N, \Gamma)$ be an acyclic transitive digraph. If G has the minimum node, then Algorithm* 1 *potentially yields all linear orderings on G.*

Proof. First, note that G is a poset by Lemma 2.1. Since G has the minimum node, Algorithm 1 yields a linear ordering on G by Lemma 4.3. Let $(i_1, i_2, \ldots, i_n)$ be an arbitrary linear ordering on G, i.e., $(i_h, i_{h'}) \in \Gamma$ implies that $h < h'$. We will show that Algorithm 1 can produce the linear ordering $(i_1, i_2, \ldots, i_n)$. Since i_1 is an undominated node of N, Algorithm 1 can choose i_1 at Step 2. Suppose that Algorithm 1 grows the tree T in the order of $i_1, i_2, \ldots, i_{h-1}$. It suffices to show that Algorithm 1 can choose i_h at the next iteration. When i_{h-1} is selected as j_k, $Q_{i_{h-1}} = N \setminus \{i_1, i_2, \ldots, i_{h-1}\}$ and $Q_j = \emptyset$ for all $j \in N \setminus \{i_{h-1}\}$. Hence Algorithm 1 choose $Q_{i_{h-1}}$ as Q_i at Step 4 and go to Step 3. At Step 3, since $Q_{i_{h-1}}$ is connected through the minimum node of G, the connected component of $Q_{i_{h-1}}$ is $Q_{i_{h-1}}$ itself. Since $(i_1, i_2, \ldots, i_n)$ is a linear ordering on G, there is no node $j \in N \setminus \{i_1, i_2, \ldots, i_{h-1}\}$ such that $(j, i_h) \in \Gamma$. By Lemma 4.1, i_h is an undominated node of $N \setminus \{i_1, i_2, \ldots, i_{h-1}\} = Q_{i_{h-1}}$, i.e., $i_h \in U_\Gamma(Q_{i_{h-1}})$. Thus, Algorithm 1 can choose i_h and set $T = T \cup \{(i_{h-1}, i_h)\}$ at the next iteration. □

5. Computational complexity of the average covering tree value

To discuss the computational complexity of the average covering tree value, we give another representation of the average covering tree value when a digraph is a poset.

Lemma 5.1. *Given a digraph game (N, v, Γ) such that the digraph $G = (N, \Gamma)$ is acyclic and transitive. Suppose that G has the minimum node. Then the average covering tree value of player $i \in N$ is rewritten as follows:*

(5.1)
$$\mathrm{ACT}_i(N, v, \Gamma) = \frac{1}{|\mathcal{R}(\Gamma)|} \sum_{\substack{S \subseteq N; \\ i \in U_\Gamma(S), \\ i \in D_\Gamma((N \setminus S) \cup \{i\})}} |\mathcal{R}(\Gamma|_{S\setminus\{i\}})| \cdot |\mathcal{R}(\Gamma|_{N\setminus S})| (v(S) - v(S \setminus \{i\})).$$

Proof. First, note that G is a poset by Lemma 2.1. Since G has the minimum node, Algorithm 1 produces all linear orderings on G by Lemma 4.4. Hence, the average covering tree value is given by

$$\mathrm{ACT}_i(N, v, \Gamma) = \frac{1}{|\mathcal{R}(\Gamma)|} \sum_{\pi \in \mathcal{R}(\Gamma)} [v(\{j \in N | \pi(i) \leq \pi(j)\}) - v(\{j \in N | \pi(i) \leq \pi(j)\} \setminus \{i\})].$$

For each $S \subseteq N$ such that $i \in U_\Gamma(S)$ and $i \in D_\Gamma((N \setminus S) \cup \{i\})$, there are

$$|\mathcal{R}(\Gamma|_{S\setminus\{i\}})| \cdot |\mathcal{R}(\Gamma|_{N\setminus S})|$$

linear orderings $\pi \in \mathcal{R}(\Gamma)$ where $S = \{\, j \in N \mid \pi(i) \leq \pi(j) \,\}$. Therefore the average covering tree value is given by the formula (5.1). □

We will prove the following theorem in a similar way to the proof of Proposition 3 in Faigle and Kern [3].

Proposition 5.2 (# P-completeness of the average covering tree value)**.** *Assume that there exists a polynomial-time algorithm to compute the average covering tree value for given digraph games. Then there exists a polynomial-time algorithm to compute the number of all linear orderings for any posets.*

Proof. Given an arbitrary poset $P = (N, \Gamma)$, we form a digraph $G^* = (N^*, \Gamma^*)$ from the poset as follows:

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

The digraph G^* is a poset, that is, an acyclic and transitive digraph. Let $(i_1, i_2, \ldots, i_{n+1})$ be any linear ordering on G^*. Consider a digraph game $(N_k^*, \delta_{i_k}, \Gamma_k^*)$ such that $N_k^* = \{i_k, i_{k+1}, \ldots, i_{n+1}\}$, $\Gamma_k^* = \Gamma|_{N_k^*}$, and

$$\delta_{i_k}(S) = \begin{cases} 1 & \text{if } S = N_k^* \\ 0 & \text{otherwise} \end{cases}$$

for $k = 1, 2, \ldots, n+1$. Note that for all $k = 1, 2, \ldots, n+1$, N_k^* is connected through $i_{n+1} = i^*$, i_k is an undominated node of N_k^*, and every (N_k^*, Γ_k^*) is a poset. Since each N_k^* contains the minimum node i^*, the formula (5.1) yields

$$\begin{aligned} &ACT_{i_k}(N_k^*, \delta_{i_k}, \Gamma_k^*) \\ &= \frac{1}{|\mathcal{R}(\Gamma_k^*)|} \sum_{\substack{S \subseteq N_k^*; \\ i_k \in U_\Gamma(S), \\ i_k \in D_\Gamma((N_k^* \setminus S) \cup \{i_k\})}} |\mathcal{R}(\Gamma^*|_{S\setminus\{i_k\}})| \cdot |\mathcal{R}(\Gamma^*|_{N_k^*\setminus S})| (\delta_{i_k}(S) - \delta_{i_k}(S \setminus \{i\})) \\ &= \frac{1}{|\mathcal{R}(\Gamma_k^*)|} |\mathcal{R}(\Gamma^*|_{N_k^*\setminus\{i_k\}})| \cdot |\mathcal{R}(\emptyset)| = \frac{|\mathcal{R}(\Gamma_{k+1}^*)|}{|\mathcal{R}(\Gamma_k^*)|} \end{aligned}$$

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

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

Since i^* is the minimum node of (N^*, Γ^*), the number of all linear orderings on N^* is equal to the number of all linear orderings on $N^* \setminus \{i^*\} = N$, i.e.,

$$|\mathcal{R}(\Gamma^*)| = |\mathcal{R}(\Gamma^*|_{N^* \setminus \{i^*\}})| = |\mathcal{R}(\Gamma^*|_N)| = |\mathcal{R}(\Gamma)|.$$

Consequently,

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

□

By Theorem (5.2), we are able to compute $|\mathcal{R}(\Gamma)|$ in polynomial-time if there is a polynomial-time algorithm to compute $ACT_{i_k}(N_k^*, \delta_{i_k}, \Gamma|_{N_k^*})$ for $k = 1, 2, \ldots, n$. Brightwell and Winkler [2], however, proved that the problem of counting the number of all linear orderings is #P-complete. Therefore, it is doubtful whether there is an efficient algorithm to compute the exact average covering tree value.

References

[1] Bondy, J.A., Murty, U.S.R. *Graph Theory*, Springer (2008).

[2] Brightwell, G., Winkler, P.: Counting linear extensions is # P-complete. *Order* (1991) 8(3): 175–181.

[3] Faigle, U., Kern, W. : The Shapley value for cooperative games under precedence constraints. *International Journal of Game Theory* (1992) 21: 249–266.

[4] 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* (2011) 203–212.

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

(Yamamoto) Faculty of Engineering, Information and Systems, University of Tsukuba, Tsukuba, Ibaraki 305-8573, Japan
E-mail address: `yamamoto@sk.tsukuba.ac.jp`