Operations Research Letters 43 (2015) 89–92

Contents lists available at ScienceDirect

Operations Research Letters

journal homepage: www.elsevier.com/locate/orl

# The nucleolus of arborescence games in directed acyclic graphs

Naoyuki Kamiyama

*Institute of Mathematics for Industry, Kyushu University, Japan*

ARTICLE INFO

*Article history:*
Received 26 February 2014
Received in revised form
7 December 2014
Accepted 7 December 2014
Available online 13 December 2014

*Keywords:*
Arborescence game
Nucleolus
Directed acyclic graph

ABSTRACT

In this paper, we consider the problem of finding the nucleolus of arborescence games. We prove that the nucleolus of arborescence games in directed acyclic graphs can be found in linear time.



## 1. Introduction

In cooperative game theory, we consider how to distribute a profit generated by a group to its members. For this problem, several solution concepts have been proposed. One of the most important solution concepts is the concept of the core. This concept requires that no subgroup of members can benefit by breaking away from the grand coalition. However, even if the core is nonempty, it may not be a singleton. Hence, it is unclear how to distribute a profit. The concept of the nucleolus was introduced by Schmeidler [22] as a singleton solution of cooperative games. The nucleolus is the unique distribution that lexicographically maximizes the vector of non-decreasingly ordered excesses. It is well known that if the core is nonempty, then it always contains the nucleolus.

In this paper, we consider the algorithmic problem of finding the nucleolus. For general cooperative games, Kopelowitz [15] and Maschler, Peleg, and Shapley [17] proposed an algorithm for finding the nucleolus based on solving a sequence of linear programs. However, the size of these linear programs may be exponentially large in the input size. Computing the nucleolus is a notoriously hard problem. The first polynomial-time algorithm computing the nucleolus was proposed by Megiddo [18] for tree enterprises. Later on, several polynomial-time algorithms were proposed [23,14,12,5,21,9,1,3,2,4,19,11,16]. On the other hand, **NP**-hardness results for computing the nucleolus were proved in the papers [5,10,8].

In this paper, we consider the problem of finding the nucleolus of arborescence games introduced by Deng, Ibaraki, and Nagamochi [6]. Arborescence games model a situation in which we want to maintain paths from a specified vertex to all vertices in a network [6]. In this paper, we prove that the nucleolus of arborescence games in directed acyclic graphs can be found in linear time.

The rest of this paper is organized as follows. In Section 2, we give necessary definitions and known results. In Section 3, we consider essential coalitions of arborescence games. In Section 4, we prove that the nucleolus of arborescence games in directed acyclic graphs can be found in linear time.

## 2. Preliminaries

We denote by $\mathbb{R}$ the set of reals. For each finite set $N$, each subset $S$ of $N$, and each vector $x$ in $\mathbb{R}^N$, we define $x(S) := \sum_{a\in S} x(a)$.

### 2.1. Cooperative games and solution concepts

A *cooperative game* is a pair $(N, v)$ of a finite set $N$ of *players* and a function $v: 2^N \to \mathbb{R}$ such that $v(\emptyset) = 0$. We call a subset of $N$ a *coalition*. A vector $x$ in $\mathbb{R}^N$ is called an *allocation* of $(N, v)$, if $x(N) = v(N)$. An allocation $x$ of $(N, v)$ is called an *imputation* of $(N, v)$, if $x(i) \geq v(\{i\})$ for every player $i$ in $N$. We denote by $\mathcal{I}(N, v)$ the set of imputations of $(N, v)$. That is, we define

$$\mathcal{I}(N, v) := \{x \in \mathbb{R}^N \mid x(N) = v(N),\ \forall i \in N: x(i) \geq v(\{i\})\}.$$

We define the *core* $C(N, v)$ of $(N, v)$ by

$$C(N, v) := \{x \in \mathbb{R}^N \mid x(N) = v(N),\ \forall S \subseteq N: x(X) \geq v(S)\}.$$

For each vector $x$ in $\mathbb{R}^N$ and each coalition $S$ of $N$, we define an *excess* $e(S, x)$ by

$$e(S, x) := x(S) - v(S).$$

*E-mail address:* kamiyama@imi.kyushu-u.ac.jp.

http://dx.doi.org/10.1016/j.orl.2014.12.005

For each vector $x$ in $\mathbb{R}^N$, the *excess vector* $\theta(x)$ is defined as the $2^{|N|} - 2$ dimensional vector whose components are the excesses for non-empty coalitions $S$ of $N$ such that $S \neq N$ and are arranged in a non-decreasing order. The *nucleolus* of $(N, v)$ is defined as the imputation $x$ of $(N, v)$ that lexicographically maximizes $\theta(x)$. It is well known [20] that if the core is nonempty, then the nucleolus always exists and it is contained in the core.

Kopelowitz [15] and Maschler, Peleg, and Shapley [17] proposed an algorithm for computing the nucleolus of $(N, v)$ by recursively solving the following linear programs $\mathbf{LP}_k$. Initially, we define $\mathcal{J}_0 := \{\emptyset, N\}$ and $\varepsilon_0 := 0$. For each positive integer $k$, the linear program $\mathbf{LP}_k$ is defined by

$$\mathbf{LP}_k \left| \begin{array}{lll} \max & \varepsilon & \\ \text{s.t.} & x(S) \geq v(S) + \varepsilon & (S \in 2^N \setminus (\mathcal{J}_0 \cup \mathcal{J}_1 \cup \cdots \cup \mathcal{J}_{k-1})) \\ & x(S) = v(S) + \varepsilon_t & (t = 0, 1, \ldots, k-1 \text{ and } X \in \mathcal{J}_t) \\ & (\varepsilon, x) \in \mathbb{R} \times \mathcal{I}(N, v). & \end{array} \right.$$

For each positive integer $k$, we define $\varepsilon_k$ as the optimum objective value of the linear program $\mathbf{LP}_k$, and we define a family $\mathcal{J}_k$ of coalitions of $N$ by

$$\mathcal{J}_k := \{S \subseteq N \mid \forall x \in \mathcal{O}_k\colon x(S) = v(S) + \varepsilon_k\},$$

where a set $\mathcal{O}_k$ of vectors in $\mathbb{R}^N$ is defined by

$$\mathcal{O}_k := \{x \in \mathbb{R}^N \mid (\varepsilon_k, x) \text{ is an optimal solution of } \mathbf{LP}_k\}.$$

This sequential linear programming process for computing the nucleolus is denoted by $\mathsf{SLP}(N, v)$. A coalition $X$ of $N$ is said to be *essential*, if

$$v(S) > \sum_{T \in \mathcal{P}} v(T)$$

for every partition $\mathcal{P}$ of $S$ such that $\mathcal{P} \neq \{S\}$. We define $\mathcal{E}(N, v)$ as the set of all essential coalitions of $N$. It is known [13] that if the core is not empty, the nucleolus can be completely determined by essential coalitions.

**Theorem 1** (*Huberman [13]*)**.** *For each cooperative game $(N, v)$ whose core is not empty, dropping the constraints for all coalitions $S$ of $N$ that is not contained in $\mathcal{E}(N, v)$ do not change the result of $\mathsf{SLP}(N, v)$.*

It is known [11] that essential coalitions form a *characterization set* for the nucleolus. Granot, Granot, and Zhu [11] proved that if the size of a characterization set is polynomially bounded in the number of players, and the identification of the elements in the characterization set can be done in strongly polynomial time, then the nucleolus can be found in strongly polynomial time. However, in arborescence games, there can be exponentially many essential coalitions (see Section 3).

### 2.2. Arborescence games

Let $D = (V, A)$ be a directed graph with a specified vertex $r$. We allow $D$ to have parallel arcs, but it has no loop. For each subset $X$ of $V$, we define

$$\varrho_D(X) := \{a = (u, w) \in A \mid u \notin X, w \in X\}.$$

That is, $\varrho_D(X)$ represents the set of arcs entering $X$. For each vertex $u$ in $V$, we write $\varrho_D(u)$ instead of $\varrho_D(\{u\})$. A subset $F$ of $A$ is called an *$r$-cut set* of $D$, if there exists a subset $X$ of $V$ such that $r \notin X$ and $\varrho_D(X) = F$. An $r$-cut set whose size is minimum among all $r$-cut sets is called a *minimum $r$-cut set*.

A subgraph $T = (V, B)$ of $D$ is called an *$r$-arborescence*, if it satisfies the following two conditions.

- For every vertex $u$ in $V$, there exists a directed path from $r$ to $u$ in $T$.
- $\varrho_D(r) \cap B = \emptyset$, and $|\varrho_D(u) \cap B| = 1$ for every vertex $u$ in $V \setminus \{r\}$.

It is not difficult to see that an $r$-arborescence is a spanning tree whose arcs are directed away from $r$. The following theorem proved by Edmonds [7] plays an important role in the sequel.

**Theorem 2** (*Edmonds [7]*)**.** *In each directed graph $D = (V, A)$ with a specified vertex $r$, there exist $k$ arc-disjoint $r$-arborescences if and only if $|F| \geq k$ for every $r$-cut set $F$ of $D$.*

We define a function $v_D\colon 2^A \to \mathbb{R}$ as follows. For each subset $B$ of $A$, $v_D(B)$ is defined as the maximum number of arc-disjoint $r$-arborescences in the directed graph $H = (V, B)$. Theorem 2 implies that if the minimum size of an $r$-cut set of $D$ is equal to $k$, then $v_D(A) = k$. An *arborescence game* in $D$ is defined as the pair $(A, v_D)$.

A cooperative game $(N, v)$ is called a *convex game*, if

$$\forall S, T \subseteq N\colon v(S) + v(T) \leq v(S \cup T) + v(S \cap T).$$

Kuipers [16] proposed a polynomial-time algorithm computing the nucleolus of convex games. Unfortunately, an arborescence game is not always a convex game. Let $D = (V, A)$ a directed acyclic graph such that

$$V = \{r, u, w\}, \qquad A = \{a = (r, u), b = (r, u), c = (u, w)\}.$$

In this graph, since

$$v_D(\{a, c\}) = 1, \qquad v_D(\{b, c\}) = 1,$$
$$v_D(\{a, b, c\}) = 1, \qquad v_D(\{c\}) = 0,$$

we have

$$v_D(\{a, c\}) + v_D(\{b, c\}) > v_D(\{a, b, c\}) + v_D(\{c\}).$$

Thus, the arborescence game in this graph is not a convex game.

An $r$-arborescence $T = (V, B)$ in $D$ is said to be *proper*, if $|F \cap B| = 1$ for every minimum $r$-cut set $F$ of $D$. An arc $a$ in $A$ is called a *dummy arc*, if

$$v_D(A) = v_D(A \setminus \{a\}),$$

i.e., there exists $v_D(A)$ arc-disjoint $r$-arborescences in the directed graph $H = (V, A \setminus \{a\})$.

**Lemma 3.** *For each directed graph $D = (V, A)$ with a specified vertex $r$ and each arc $a$ in $A$, $a$ is a dummy arc if and only if there exists no minimum $r$-cut set $F$ of $D$ such that $a \in F$.*

**Proof.** This lemma immediately follows from Theorem 2. □

The following result related to the core of arborescence games is known.

**Theorem 4** (*Deng, Ibaraki, and Nagamochi [6]*)**.** *For every directed graph $D = (V, A)$ with a specified vertex $r$, $C(A, v_D)$ is not empty.*

## 3. Essential coalitions of arborescence games

In this section, we consider essential coalitions of arborescence games. In the sequel, we call a subset of an arc set a coalition.

**Lemma 5.** *For each directed graph $D = (V, A)$ with a specified vertex $r$ and each non-empty coalition $B$ of $A$, $B \in \mathcal{E}(A, v_D)$ if and only if $B$ consists of a single arc in $A$ or the directed graph $T = (V, B)$ is an $r$-arborescence.*

**Proof.** We first prove the *if* part. For every arc $a$ in $A$, there exists no partition $\mathcal{P}$ of $\{a\}$ such that $\mathcal{P} \neq \{\{a\}\}$. This implies that $\{a\}$ is essential. Assume that we are given an $r$-arborescence $T = (V, B)$. For every coalition $F$ of $B$ such that $F \neq B$, since the directed graph $H = (V, F)$ does not contain an $r$-arborescence, $v_D(F) = 0$ holds. Since $v_D(B) = 1$, this observation implies that $B$ is essential.

Next we prove the *only if* part. Assume that $B$ is a non-empty coalition in $\mathcal{E}(A, v_D)$. If $B$ contains more than one arc and the directed graph $T = (V, B)$ does not contain an $r$-arborescence, then $v_D(F) = 0$ for every coalition $F$ of $B$. This implies that $B$ is not essential. We assume that the directed graph $H = (V, B)$