Strongly Essential Coalitions and the Nucleolus of Peer Group Games

Brânzei, R.; Solymosi, T.; Tijs, S.H.

Publication date:
2003

Citation for published version (APA):
Brânzei, R., Solymosi, T., & Tijs, S. H. (2003). *Strongly Essential Coalitions and the Nucleolus of Peer Group Games*. (CentER Discussion Paper; Vol. 2003-19). Microeconomics.

No. 2003–19

STRONGLY ESSENTIAL COALITIONS AND THE NUCLEOLUS OF PEER GROUP GAMES

By Rodica Brânzei, Tamás Solymosi and Stef Tijs

February 2003

ISSN 0924-7815

Strongly essential coalitions and the nucleolus of peer group games [*]

Rodica Brânzei[†] Tamás Solymosi[‡] Stef Tijs[§]

February 24, 2003

Abstract

Most of the known efficient algorithms designed to compute the nucleolus for special classes of balanced games are based on two facts: (i) in any balanced game, the coalitions which actually determine the nucleolus are essential; and (ii) all essential coalitions in any of the games in the class belong to a prespecified collection of size polynomial in the number of players.

We consider a subclass of essential coalitions, called strongly essential coalitions, and show that in any game, the collection of strongly essential coalitions contains all the coalitions which actually determine the core, and in case the core is not empty, the nucleolus and the kernelcore.

As an application, we consider peer group games, and show that they admit at most $2n-1$ strongly essential coalitions, whereas the number of essential coalitions could be as much as 2^{n-1}. We propose an algorithm that computes the nucleolus of an n-player peer group game in $\mathcal{O}(n^2)$ time directly from the data of the underlying peer group situation.

JEL classification: C71
Keywords: cooperative game, peer group game, nucleolus, kernel, computation

[*]Research supported in part by OTKA grant T030945.

[†]Faculty of Computer Science, "Al.I. Cuza" University, 11, Carol I Bd., 6600 Iaşi, Romania. E-mail: branzeir@infoiasi.ro

[‡]Department of Operations Research, Budapest University of Economic Sciences and Public Administration, 1828 Budapest, Pf. 489, Hungary. E-mail: tamas.solymosi@bkae.hu

[§]CentER and Department of Econometrics and Operations Research, Tilburg University, P.O. Box 90153, 5000 LE Tilburg, The Netherlands. E-mail: S.H.Tijs@uvt.nl

1 Introduction

The nucleolus is a well-established solution concept for transferable utility cooperative games. It seemingly depends on all coalitional values, but a closer look reveals the inherent high redundancy. Indeed, as Brune (1983), and more recently Reijnierse and Potters (1998) have proved: in any n-player game there are at most $2n-2$ coalitions which actually determine the nucleolus. Unfortunately, the identification of these coalitions is no less laborious as computing the nucleolus itself. On the other hand, if special properties of the game allow us to specify a priori a collection of polynomial size that contains all nucleolus-defining coalitions, then (and only then) we can compute the nucleolus in time polynomial in the number of players.

Many classes of games for which efficient nucleolus algorithms are available have the following properties:

- there is a collection of polynomial size which contains all essential coalitions in any of the games in that class, and
- the core is nonempty in any of the games in the class.

The key to the efficiency of these algorithms is the useful result of Huberman (1980): in balanced games all nucleolus-defining coalitions are essential.

We identify a subclass of essential coalitions, called strongly essential coalitions, with an increased simplification potential. We show that in any game, the collection of strongly essential coalitions contains all the coalitions which actually determine the core, and in case the core is nonempty, the nucleolus. Using the terminology of Granot et al. (1998): in balanced games the collection of strongly essential coalitions is a characterization set for the nucleolus. Compared to the concept of essentiality, the added value in strong essentiality is that coalitions must prove to be non-redundant even against families of non-disjoint coalitions. Bondareva and Driessen (1994) used this idea to determine exact bounds for individual payoffs in the core.

The reduction potential of using strong essentiality in the computation of the nucleolus can be well illustrated by the class of cyclic permutation games (Solymosi et al., 2000), a large subclass of permutation games which contains assignment games as well as permutation games whose grand coalition is essential. Indeed, it can happen in an n-player cyclic permutation game that all the 2^n coalitions survive a redundancy check with disjoint families (not in an assignment game, of course), however, with overlapping families all but at most $n(n-1)$ coalitions are filtered out (like in an assignment game).

As another application, in the second part of this paper we consider peer group games, and identify a collection of $2n-1$ coalitions that contains all strongly essential coalitions. In contrast, the number of essential coalitions could be as much as $2^{n-1}+n-1$. We propose an algorithm that computes the nucleolus of an n-player peer group game in $\mathcal{O}(n^2)$ time. It works directly from the data of the underlying peer group situation, so the coalition values need not be explicitly calculated.

The outline of the rest of the paper is as follows. We recall the necessary definitions in the next section. Strongly essential coalitions are introduced and their simplification possibilities for excess-based solutions in general games are discussed in section 3. We specialize the general results to peer group games in section 4. Section 5 contains an $\mathcal{O}(n^2)$ algorithm for the nucleolus of an n-player peer group game. We illustrate the algorithm on a 9-player game in section 6.

2 Preliminaries

We start with the necessary definitions and notations. Let (N, v) be a transferable utility cooperative game, where N is the nonempty, finite set of *players* and $v : 2^N \to \mathbb{R}$ is the *coalitional function* satisfying $v(\varnothing) = 0$. We use the standard notation $x(S) := \sum_{i \in N} a_i^S x_i$ for the total payoff to *coalition* $S \subseteq N$ at *payoff allocation* $x \in \mathbb{R}^N$, where $a^S \in \mathbb{R}^N$ denotes the membership vector of coalition S (i.e., $a_i^S = 1$ if $i \in S$, $a_i^S = 0$ if $i \notin S$).

Given a game (N, v), the *excess* $e(S, x) := v(S) - x(S)$ of a coalition S at a payoff allocation x is the usual measure of gain (or loss if negative) to S, if its members depart from the proposed x in order to form their own coalition. Note that $e(\varnothing, x) = 0$ for all $x \in \mathbb{R}^N$. A payoff allocation $x \in \mathbb{R}^N$ is called *efficient*, if $e(N, x) = 0$; *individually rational*, if $e(\{i\}, x) \leq 0$ for all $i \in N$; and *coalitionally rational*, if $e(S, x) \leq 0$ for all $S \subseteq N$. We denote by $\mathcal{I}^=$ the *preimputation set* (i.e., the set of efficient payoffs), by $\mathcal{I}$ the *imputation set* (i.e., the set of individually rational preimputations), and by $\mathcal{C}$ the *core* (i.e., the set of efficient and coalitionally rational payoffs).

The *kernel*, denoted here by $\mathcal{K}$, was introduced by Davis and Maschler (1965). In this paper we only consider its intersection with the core. We call it *kernelcore*, and denote it by $\mathcal{KC}$. It was shown by Maschler et al. (1979) that for any game, $\mathcal{KC} = \mathcal{K} \cap \mathcal{C} = \{x \in \mathcal{C} : s_{ij}(x) = s_{ji}(x) \;\; \forall\, i \neq j\}$, where the function $s_{ij}(x) := \max\{e(S, x) : S \subset N,\, S \ni i,\, S \not\ni j\}$ measures the *surplus* of i against j at x.

The *nucleolus*, denoted here by $\mathcal{N}$, was introduced by Schmeidler (1969). It is a subset of the kernel (and in case the core is not empty, of the kernelcore) that consists of a single payoff allocation, called the *nucleolus allocation*. The nucleolus is defined as follows. For payoff allocation x let $E(x) = [\ldots \geq e(S, x) \geq \ldots : \varnothing \neq S \neq N]$ denote the $(2^n - 2)$-component vector that is composed of the nonincreasingly arranged excesses of proper coalitions at x. The nucleolus is the set of imputations which lexicographically minimize the vector-valued function $E(.)$ over the imputation set. Formally, $\mathcal{N} = \{x \in \mathcal{I} : E(x) \preceq_{lex} E(y) \;\; \forall\, y \in \mathcal{I}\}$, where $\preceq_{lex}$ denotes the lexicographic order of vectors.

One obtains generalizations of the core, the kernelcore, and the nucleolus, by specifying a nonempty collection $\mathcal{B} \subseteq 2^N \setminus \{\varnothing, N\}$ of coalitions, and by requiring that only the coalitions in $\mathcal{B} \cup \{N\}$ determine the solution. We define the *$\mathcal{B}$-core* as $\mathcal{C}^{\mathcal{B}} = \{x \in \mathbb{R}^N : e(N, x) = 0;\;\; e(S, x) \leq 0 \;\; \forall\, S \in \mathcal{B}\}$; the *$\mathcal{B}$-kernelcore* as $\mathcal{KC}^{\mathcal{B}} = \{x \in \mathcal{C}^{\mathcal{B}} : s_{ij}^{\mathcal{B}}(x) = s_{ji}^{\mathcal{B}}(x) \;\; \forall\, i \neq j\}$, where $s_{ij}^{\mathcal{B}}(x) := \max\{e(S, x) : S \in \mathcal{B},\, S \ni i,\, S \not\ni j\}$ measures the *$\mathcal{B}$-surplus* of i against j at x; and the *$\mathcal{B}$-nucleolus* as $\mathcal{N}^{\mathcal{B}} = \{x \in \mathcal{I} : E^{\mathcal{B}}(x) \preceq_{lex} E^{\mathcal{B}}(y) \;\; \forall\, y \in \mathcal{I}\}$, where $E^{\mathcal{B}}(x) = [\ldots \geq e(S, x) \geq \ldots : S \in \mathcal{B}]$ is the $|\mathcal{B}|$-component vector of the nonincreasingly arranged excesses of the coalitions in $\mathcal{B}$ at x.

Such generalizations might be useful in modeling situations where the cooperation of the players is restricted by some external factor (e.g. by communication difficulties), and so it is natural to demand that only the coalitions which are capable of forming influence the suggested outcome(s). For this paper, a more important reason is that the core, kernelcore, and nucleolus are often determined by a small collection of coalitions, and hence, the computation of the restricted solutions is significantly easier. For such a collection a well-known example is the collection of essential coalitions.

In a game (N, v), a coalition S is called *inessential* if it has a proper partition $S = \{S_1, \ldots, S_r\}$, $r \geq 2$, such that $v(S) \leq \sum_{j=1}^{r} v(S_j)$, consequently, $e(S, x) \leq \sum_{j=1}^{r} e(S_j, x)$ for all $x \in \mathbb{R}^N$. Coalitions which are not inessential are called *essential.* Notice that single-player coalitions are always essential. The core is obviously independent of inessential coalitions. It was noticed by Huberman (1980) that in case the core is not empty, also the nucleolus is independent of inessential coalitions. By an analogous argument one easily shows that in case the core is not empty, the kernelcore is independent of inessential coalitions. More precisely, if a collection $\mathcal{B} \cup \{N\}$ contains all essential coalitions in an n-player game, we have

- $rank\{a^B : B \in \mathcal{B} \cup \{N\}\} = n$;
- $\mathcal{C}^{\mathcal{B}} = \mathcal{C}$;
- if $\mathcal{C}^{\mathcal{B}} \neq \varnothing$ then $\mathcal{KC}^{\mathcal{B}} = \mathcal{KC}$ and $\mathcal{N}^{\mathcal{B}} = \mathcal{N}$.

In the following section we identify a subclass of essential coalitions with the same properties.

3 Strongly essential coalitions

In this section we consider special essential coalitions, which we call strongly essential coalitions, and show that in any game the collection of strongly essential coalitions contains all the coalitions which actually determine the core, and in case the core is nonempty, the kernelcore and the nucleolus.

In a game (N, v), we say that a coalition S is *weakly inessential* if there are coalitions $\{S_j\}_{j=1}^{r}$, $r \geq 2$, and numbers $\mu, \sigma \geq 0$ such that

$$a^S = \sum_{j=1}^{r} a^{S_j} - \mu\, a^N \qquad \text{and} \qquad v(S) = \sum_{j=1}^{r} v(S_j) - \mu\, v(N) - \sigma. \tag{1}$$

The key feature of such a weakly inessential S is that

$$e(S, x) = \sum_{j=1}^{r} e(S_j, x) - \sigma \qquad \text{whenever } x \text{ is efficient.} \tag{2}$$

Obviously, an inessential coalition is weakly inessential with $\mu = 0$. Coalitions which are not weakly inessential are called *strongly essential.* By an inductive argument it is easily seen that *any weakly inessential S has a decomposition (1) such that all $S_j \in \mathcal{B}$ $(j = 1, \ldots, r)$ are strongly essential.*

As an illustration, consider the $n(\geq 3)$-player symmetric game with coalitional function $v(S) = |S| - 1$ for $S \neq \varnothing$. All coalitions are essential, but only the $(n-1)$- and n-player coalitions are strongly essential. Indeed, for a coalition S with $1 \leq |S| \leq n-2$ and for a proper partition of $N \setminus S = T_1 \cup T_2$ we have $v(S) = v(S \cup T_1) + v(S \cup T_2) - v(N)$, showing the weak inessentiality of S.

Next we summarize the properties of strongly essential coalitions which are useful in the computation of excess-based solutions.

Theorem 1 *Let a collection $\mathcal{B} \cup \{N\}$ contain all strongly essential coalitions in an n-player game. Then*
(i) $rank\{a^B : B \in \mathcal{B} \cup \{N\}\} = n$;
(ii) $\mathcal{C}^{\mathcal{B}} = \mathcal{C}$;
(iii) *if* $\mathcal{C}^{\mathcal{B}} \neq \varnothing$ *then* $\mathcal{KC}^{\mathcal{B}} = \mathcal{KC}$;
(iv) *if* $\mathcal{C}^{\mathcal{B}} \neq \varnothing$ *then* $\mathcal{N}^{\mathcal{B}} = \mathcal{N}$.

Proof
(i) If $\{i\} \notin \mathcal{B}$ then $\{i\}$ is weakly inessential, so its membership vector $a^{\{i\}}$ is a linear combination of the membership vectors a^B $(B \in \mathcal{B} \cup \{N\})$. Since $rank\{a^{\{i\}} : i \in N\} = n$, the claim follows.

(ii) Only $\mathcal{C}^{\mathcal{B}} \subseteq \mathcal{C}$ needs explanation. Take an $S \notin \mathcal{B}$. It has a decomposition (1) such that $S_j \in \mathcal{B}$ for all $j = 1, \ldots, r$. If $x \in \mathcal{C}^{\mathcal{B}}$ then $e(S_j, x) \leq 0$ for all $j = 1, \ldots, r$, so from (2) we get $e(S, x) \leq 0$, i.e., S is redundant for $\mathcal{C}$. It follows that $\mathcal{C}^{\mathcal{B}} \subseteq \mathcal{C}$.

(iii) We only need to show that for any pair of players $i \neq j$, $s_{ij}^{\mathcal{B}}(x) = s_{ij}(x)$ holds for all $x \in \mathcal{C}$. Obviously, $s_{ij}^{\mathcal{B}}(x) \leq s_{ij}(x)$ for all $x \in \mathbb{R}^N$.

On the other hand, at an $x \in \mathcal{C}$, let $s_{ij}(x) = e(S, x)$ for some coalition S. If S is in $\mathcal{B}$, we get $e(S, x) \leq s_{ij}^{\mathcal{B}}(x)$, and the claim follows. If S is not in $\mathcal{B}$, there is a collection $\{S_k\}_{k=1}^{r}$ and a real number μ such that $a^S = \sum_{k=1}^{r} a^{S_k} - \mu\, a^N$ and $v(S) \leq \sum_{k=1}^{r} v(S_k) - \mu\, v(N)$. Since x is efficient, $e(S, x) \leq \sum_{k=1}^{r} e(S_k, x)$. Clearly, we can assume without loss of generality that $S_k \in \mathcal{B}$ for all $k = 1, \ldots, r$. It follows from the nonpositivity of the excesses at core allocations that $e(S, x) \leq e(S_k, x)$ for all $k = 1, \ldots, r$. Moreover, there must be at least one coalition, say S_k, in the decomposing collection which contains i but not j. We get $e(S, x) \leq e(S_k, x) \leq s_{ij}^{\mathcal{B}}(x)$, and the claim follows.

(iv) It follows from *(i)* that $\mathcal{N}^{\mathcal{B}}$ is a singleton, so we only need to show that $E^{\mathcal{B}}(x) \preceq_{lex} E^{\mathcal{B}}(y)$ implies $E(x) \preceq_{lex} E(y)$ for any $x, y \in \mathcal{C}$. Intuitively it is clear: in the excess-profile $E(x)$ of a core allocation x, the excess of a coalition $S \notin \mathcal{B}$ is behind all the excesses in its excess-decomposition (2), so $e(S, x)$ plays no role in the lexicographic minimization until any of the excesses $e(S_k, x)$ $(k = 1, \ldots, r)$ has some role, but when all of them become fixed, so does $e(S, x)$, and S has no chance to play a role.

A formal direct proof could be obtained by repeating verbatim the proof of Huberman (1980). We omit any further details because the general results of Granot et al. (1998), or of Reijnierse and Potters (1998) easily give claim *(iv)*. $\square$

4 Peer group games

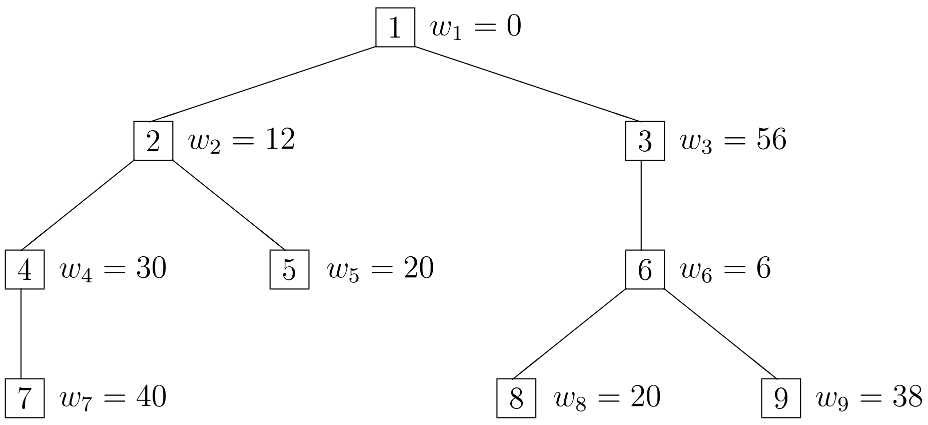
Tree-connected peer group situations and related peer group games were introduced by Brânzei et al. (2002) to model economic and operations research situations in which agents with potential individual possibilities are connected via a hierarchy within an organization.

Given $N = \{1, \ldots, n\}$, let T be a tree with node set N. We designate node 1 as the root of the tree, and hence, induce a partial ordering $\preceq$ on N. We write $j \preceq i$ if node j lies on the path from root 1 to node i. For each $i \in N$, let

$$P(i) := \{j \in N : j \preceq i\} \qquad Q(i) := \{j \in N : i \preceq j\} \qquad R(i) := N \setminus Q(i).$$

In particular, $P(1) = \{1\}$, $Q(1) = N$, and $R(1) = \varnothing$. Notice that for $i \geq 2$ the set $R(i)$ consists of the nodes of a 1-rooted subtree whose complement $Q(i)$ is an i-rooted subtree.

The set N and the rooted tree T together with a nonnegative vector $w \in \mathbb{R}^N_+$ describes a peer group situation. The following example will be used throughout the paper.



A peer group situation

With peer group situation $\langle N, T, w\rangle$ we associate a transferable utility cooperative game, a peer group game. Its set of players is N, its coalitional function v is given by

$$v(S) = \sum_{i:P(i)\subseteq S} w_i \qquad \forall\, S \subseteq N,$$

with the standard convention that summing over the empty set gives 0. Throughout the paper we assume that $w_1 = 0$, i.e., the peer group game is 0-normalized.

Given a 1-rooted tree T on node set N, let $\mathcal{B}_1$ denote the collection of coalitions corresponding to the 1-rooted subtrees of T, including the tree T itself. Further, let $\mathcal{B}_0 := \{\{j\} : j \in N \setminus \{1\}\}$, and $\mathcal{B}^* := \mathcal{B}_1 \cup \mathcal{B}_0$. We call a partition $\{S_1, \ldots, S_r\}$ of a nonempty coalition S *the $\mathcal{B}^*$-partition of S* if (i) $S_k \in \mathcal{B}^*$ for every $k = 1, \ldots, r$; (ii) the partition is of minimum size among the partitions satisfying (i). Note that the $\mathcal{B}^*$-partition exists and is unique for all nonempty $S \subseteq N$.

Clearly, the coalitional function v of the peer group game corresponding to the peer group situation $\langle N, T, w\rangle$ satisfies

$$v(S) = \begin{cases} \sum_{i\in S} w_i & \text{if } \ S \in \mathcal{B}_1 \\ 0 & \text{if } \ S \in \mathcal{B}_0 \\ \sum_{j=1}^{r} v(S_j) & \text{if } \ S \notin \mathcal{B}^* \ \text{ and } \ \{S_1, \ldots, S_r\} \text{ is the } \mathcal{B}^*\text{-partition of } S. \end{cases}$$

This implies that any 0-normalized peer-group game defined on the tree T is a T-component additive game, hence has a nonempty core (Le Breton et al., 1992; Potters and Reijnierse, 1995). Moreover, the collection $\mathcal{B}^*$ contains all essential coalitions in any peer group game defined on T. Observe that $|\mathcal{B}^*|$ can be as large as $2^{|N|-1} + |N| - 1$. Indeed, if T is the 1-rooted star-graph (i.e., each edge is of the form $\{1, j\}$ for $j \geq 2$), then $|\mathcal{B}_1| = 2^{|N|-1}$.

Let $\mathcal{B}_2$ be the collection of the node sets of the 1-rooted subtrees with nonempty connected complement, i.e., $\mathcal{B}_2 := \{R(i) : i \in N \setminus \{1\}\}$. Define $\mathcal{B} := \mathcal{B}_2 \cup \mathcal{B}_0$. We claim that the collection $\mathcal{B} \cup \{N\}$ contains all strongly essential coalitions in any peer group game defined on the underlying 1-rooted tree. To see this, let $S \in \mathcal{B}_1 \setminus (\mathcal{B}_2 \cup \{N\})$. Then $r \geq 2$ edges of the tree go out from S, let $i_1, \ldots, i_r \in N \setminus S$ be the endnodes of those edges. Since $S \cup Q(i_1) \cup \ldots \cup Q(i_r)$ is a partition of N, we have $a^S = a^{R(i_1)} + \ldots + a^{R(i_r)} - (r-1)a^N$. The coalitional function on $\mathcal{B}_1$ is the additive set function induced by the weight vector w, so $v(S) = v(R(i_1)) + \ldots + v(R(i_r)) - (r-1)v(N)$, thus S is weakly inessential. It follows that $\mathcal{B} \cup \{N\}$ indeed contains all strongly essential coalitions. Observe that independently of the structure of the tree, $|\mathcal{B}| = 2|N| - 2$.

For illustration, let $S = \{1\}$ in our 9-player example. It has $r = 2$ neighbors in the tree: $i_1 = 2$ and $i_2 = 3$. From $a^{\{1\}} = a^{\{1,2,4,5,7\}} + a^{\{1,3,6,8,9\}} - a^N$ and $v(\{1\}) = 0 = 102 + 120 - 222 = v(\{1,2,4,5,7\}) + v(\{1,3,6,8,9\}) - v(N)$ we get that coalition $\{1\}$ is weakly inessential. Note that the same is true in general whenever at least two branches start from the root. On the other hand, in case only one edge, say $\{1, 2\}$, leaves the root, we have $\{1\} = R(2)$.

Therefore, Theorem 1 applied to peer group games immediately gives

Corollary 1 *Given a 1-rooted tree T on node set $N = \{1, \ldots, n\}$, let $\mathcal{B} := \{\{i\} : i \geq 2\} \cup \{R(i) : i \geq 2\}$. Then*
(i) $\mathcal{B} \cup \{N\}$ *contains all strongly essential coalitions in any peer group game on T;*
(ii) $\mathcal{C}^{\mathcal{B}} = \mathcal{C} \neq \varnothing$;
(iii) $\mathcal{KC}^{\mathcal{B}} = \mathcal{KC}$;
(iv) $\mathcal{N}^{\mathcal{B}} = \mathcal{N}$.

In what follows we will be mainly concerned with how to utilize the last equivalence.

Since our collection $\mathcal{B}$ consists of (in the number of players) linearly many coalitions, we know from Theorem 4.8 of Granot et al. (1998) that the $\mathcal{B}$-nucleolus can be calculated in strongly polynomial time. More explicitly, due to the connectedness of each member of $\mathcal{B}$ in the underlying tree and the balancedness of the game, we can invoke the algorithm of Kuipers et al. (2000) and compute the $\mathcal{B}$-nucleolus in $\mathcal{O}(n^3|\mathcal{B}|) = \mathcal{O}(n^4)$ time. To input that algorithm, first we would need to calculate the values of the coalitions in $\mathcal{B}$ from the peer group situation. Thanks to the additive nature of the required coalitional values, such preparation would take only $\mathcal{O}(n^2)$ time.

We can do better, if we exploit the fact that player 1 is a veto player in any peer group game defined on any 1-rooted tree. Indeed, the streamlined version of the algorithm of Kuipers et al. (2000, p. 557) would compute the $\mathcal{B}$-nucleolus in $\mathcal{O}(n^2|\mathcal{B}|) = \mathcal{O}(n^3)$ time (after the coalitional values are derived in $\mathcal{O}(n^2)$ time). That variant however assumes the underlying tree to be a star-graph whose root is the veto-player, so the hierarchy among the other players as described by the peer group situation would not be utilized.

Alternatively, the nucleolus allocation of a peer group game can be determined as the unique kernel element. Since 0-normalized peer group games are tree-component additive games, the results of Potters and Reijnierse (1995) imply that the kernel consists only of the nucleolus allocation, that is the unique payoff vector satisfying efficiency and the $n-1$ kernel conditions corresponding to the pairs of players connected by an edge of the underlying tree. Brânzei et al. (2000) compute the nucleolus of line-graph peer group games by solving that system of n equations. It is possible to efficiently solve that nonlinear system even for general peer group games, in fact, the algorithm in the next section can be interpreted as doing just that.

For veto-rich games (i.e., nonnegative games with a veto player) Arin and Feltkamp (1997) proves that the nucleolus allocation can be calculated as the unique kernel element, and presents an algorithm that sequentially determines (even if the game is not balanced) the unique payoff vector satisfying efficiency and the $n-1$ kernel conditions corresponding to the veto–nonveto pairs. Since peer group games are monotonic (hence balanced) veto-rich games, in light of Corollary 1(*iii*) a specialized variant of that algorithm could be relevant for us. Indeed, if we streamline the algorithm of Arin and Feltkamp (1997) in a straightforward way for monotonic and $\mathcal{B}$-restricted veto-rich games, we basically get the specialized algorithm of Kuipers et al. (2000, p. 557). Thus, applied to peer group games we would obtain the same $\mathcal{O}(n^3)$ algorithm with the mentioned imperfection.

In the next section we propose a refinement of these algorithms that also exploits the additive nature of the coalitional values in the collection $\mathcal{B}$. This enables us to work directly on the tree and with the weights of the underlying peer group situation, and compute the nucleolus allocation in $\mathcal{O}(n^2)$ time.

5 An algorithm for the nucleolus

We present an algorithm that computes the nucleolus of a peer group game directly from the data of the underlying peer group situation.

Algorithm

```
INPUT: a peer group situation $\langle N, T, w\rangle$ with 1-rooted tree $T$ and $w_1 = 0$
Initially,
    (I1) set      $\alpha := 0,\ \ A := N \setminus \{1\}$
    (I2) compute  $p_i := w(Q(i))\ \ \forall\, i \in A$
```

```
While $A \neq \varnothing$
    (L0)  compute  $q_i := 1 + |Q(i) \cap A| \quad \forall i \in A$
    (L1)  compute  $\beta := \min\{\frac{p_i}{q_i} : i \in A\}, \quad B := \arg\min\{\frac{p_i}{q_i} : i \in A\}$
    (L2)  find     $B := A \cap Q(B)$   where $Q(B) = \cup_{j \in B} Q(j)$
    (L3)  set      $\alpha := \alpha + \beta, \quad A := A \setminus B$
    (L4)  update   $p_j := \alpha \;\; \forall j \in B, \quad p_i := p_i - \beta q_i \;\; \forall i \in A$
Finally,  set $p_1 := \sum_{i=1}^{n} w_i - \sum_{j=2}^{n} p_j$
OUTPUT: $(p_1, \ldots, p_n)$ = the nucleolus allocation of the associated peer group game
```

In light of Corollary 1(iv), the validity of the algorithm is justified if we show that the output vector is the $\mathcal{B}$-nucleolus allocation x^* of the induced peer group game. It follows from the general Kohlberg-type characterization of $\mathcal{B}$-nucleolus allocations (Maschler et al., 1992, Theorem 7.2) that the collection $\mathcal{B}_t(x^*) = \{S \in \mathcal{B} : e(S,x) \geq t\}$ is balanced or empty for any number t. It is easily checked (or implied by Proposition 3.1 of Le Breton et al., 1992) that in the setting of Corollary 1, each balanced $\mathcal{B}$-collection (i.e., all members are in $\mathcal{B}$) that is minimal (for inclusion) is a $\mathcal{B}$-partition of N, hence of the form $\mathcal{P}_i := \{R(i)\} \cup \{\{j\} : j \in Q(i)\}$ for some $i \geq 2$. We obtain that at the $\mathcal{B}$-nucleolus allocation x^*, any nonempty collection $\mathcal{B}_t(x^*)$ is the union of some of the partitions $\mathcal{P}_i$, $i \geq 2$.

We can sequentially determine the inclusively increasing sequence of the nonempty balanced collections $\mathcal{B}_t(x^*)$ by reason of the following general fact: *the sum of the excesses of coalitions in any balanced collection is the same constant for all efficient payoff allocations.* Stated in our setting for minimal balanced collections, we have that for each partition $\mathcal{P}_i$ $(i \geq 2)$ and for all efficient x,

$$\sum_{S \in \mathcal{P}_i} e(S,x) = v(R(i)) - x(R(i)) + \sum_{j \in Q(i)} (0 - x_j) = w(R(i)) - x(N) = -w(Q(i)).$$

Thus, after the initial phase of the algorithm, for each $i \geq 2$, the value $-p_i$ equals the constant total excess of $\mathcal{P}_i$. Each node $i \geq 2$ is considered *active*, because in the associated partition $\mathcal{P}_i$ at least one (initially each) coalition is *active* (i.e., its excess at x^* is not yet found). Notice that node (player) 1 is ignored, because coalition $\{1\}$ is either weakly inessential (when more than one edges leave the root), or $\{1\} = R(2)$ (when $\{1,2\}$ is the only edge incident to the root). Since the core is not empty, an excess of at most $-\alpha = 0$ can be guaranteed to all (active) coalitions in $\mathcal{B}$. Equivalently, by $e(\{i\},x) = -x_i$, a payoff of at least α can be secured for all (active) players.

In the first round of the iterative phase of the algorithm, the value $-\beta$ equals the highest excess at x^*. Indeed, $\mathcal{B}$ is the union of the partitions $\mathcal{P}_i$ $(i \geq 2)$, and in each partition $\mathcal{P}_i$ the maximum excess is minimized, if its total $-p_i = -w(Q(i))$ excess is distributed equally among its $q_i = 1 + |Q(i)|$ active members. The collection of the highest-excess coalitions at x^* is the union of the partitions $\mathcal{P}_i$ associated with the nodes $i \in B$ as found in step *(L1)* of the loop. Step *(L2)* formalizes the fact that fixing the excesses in one partition might also determine the excesses in another partition. Indeed, if $i \in B$ and $j \in Q(i)$ then it follows from $Q(j) \subseteq Q(i)$ that node j too becomes *settled*, because each coalition in the associated partition becomes *settled* (i.e., its excess at x^* has been found). In step *(L3)* the active–settled status of the nodes (players) and the guaranteed payoff to the active players are adjusted. After *(L3)*, $-\alpha$ is the smallest excess level that can be guaranteed to all coalitions which entered the round as active. The p values of these nodes are updated in step *(L4)*. The nodes in B have just become settled, they get the current guaranteed level α, the nucleolus payoff to the associated players. The nodes in A remain active, their updated p value is the negative of what is left from the total excess of the associated partition after each active coalition in that round received the increment β found in that round.

Any subsequent round starts with counting the number of still active coalitions in each active partition. Thus, after step *(L0)* we always have that

- the nodes (players) in A are the active ones;
- for each $i \in A$, the value $-p_i$ is the total excess available for distribution among the q_i active coalitions in $\mathcal{P}_i$;
- the excess of each currently active coalition at x^* is at most $-\alpha$, or equivalently, the nucleolus payoff to each active player is at least α.

The discussion above of what happens in the first round can therefore be repeated, mutatis mutandis.

The iterative phase ends when all non-root nodes become settled. Then for all $i \geq 2$, the final value of p_i is the nucleolus payoff to player i. Thus, p_1 as computed by efficiency in the final step of the algorithm is the nucleolus payoff to player 1. Therefore, the output vector $(p_1, \ldots, p_n)$ is indeed the nucleolus allocation.

We recommend to follow the worked out 9-player example in the next section. It is crafted so that all subtleties of the algorithm could be demonstrated.

Now let us see how long it takes to actually perform the steps of the algorithm. We assume that the peer group situation is given by two arrays of size n, both indexed by the node labels. The first array contains for each non-root node the other endpoint of the edge toward the root, the second array specifies the node weights. We assume that the edges are $\{i, t(i)\}$, $i \geq 2$, and the nodes are labeled (e.g. by breadth-first search) so that $t(i) < i$ for each edge. For example, our 9-player peer group situation would be given by the arrays:

i	1	2	3	4	5	6	7	8	9
$t(i)$	–	1	1	2	2	3	4	6	6
$w(i)$	0	12	56	30	20	6	40	20	38

The initialization is performed only once. Step *(I1)* is done in $\mathcal{O}(n)$ elementary operations, if, for example, the active-settled status of the nodes are represented by a third array of size n. Step *(I2)* can also be done in $\mathcal{O}(n)$ elementary operations, for example with the following backward tracking routine:

```
(I2)  for i:=n to 2
          | $p(i) := w(i)$
      for i:=n to 2
          | $p(t(i)) := p(t(i)) + p(i)$
```

Therefore, the initialization phase of the algorithm can be performed in $\mathcal{O}(n)$ time.

The iterative phase consists of at most $n-1$ rounds, because in each round at least one of the non-root nodes becomes settled.
Step *(L0)* of the loop can be done in $\mathcal{O}(n)$ elementary operations, for example with the following backward tracking routine:

```
(L0)  for i:=n to 2
          | $q(i) := 1$
      for i:=n to 2
          | if $i \in A$ then $q(t(i)) := q(t(i)) + q(i)$, $q(i) := 1 + q(i)$
```

Step *(L1)* can be done in $\mathcal{O}(n)$ elementary operations with the following forward tracking routine (M is a sufficiently large real number):

```
(L1)  set $\beta := M$, $B := \varnothing$
      for i:=1 to n
          | if $i \in A$ and $\frac{p_i}{q_i} \leq \beta$ then $\beta := \frac{p_i}{q_i}$, $B := B \cup \{i\}$
```

Step *(L2)* can also be done in $\mathcal{O}(n)$ time with the following routine:

```
(L2)  for i:=2 to n
          | if $i \in A$ and $t(i) \in B$ then $B := B \cup \{i\}$
```

Step *(L3)* requires $\mathcal{O}(n)$ time. Step *(L4)* can also be done in $\mathcal{O}(n)$ time, for example with the following routine:

```
(L4)  for i:=2 to n
          | if $i \in A$ then $p_i := p_i - \beta q_i$
          | if $i \in B$ then $p_i := \alpha$
```

Therefore, one round of the loop takes $\mathcal{O}(n)$ time, so the iterative phase can be performed in $\mathcal{O}(n^2)$ time.

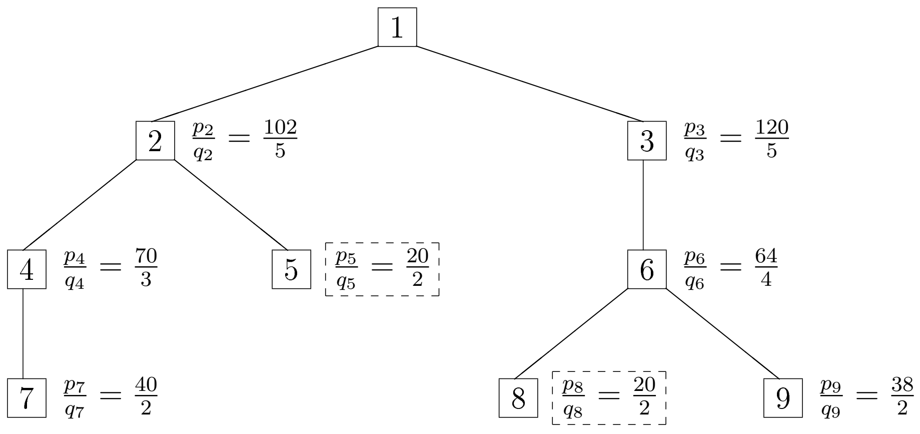
The final step requires $\mathcal{O}(n)$ time. It follows that the algorithm itself can be performed in $\mathcal{O}(n^2)$ time.

The space requirement of the algorithm is $\mathcal{O}(n)$. More explicitly, only an array of n records (indexed by the node labels) are needed, each record consisting of three types of information: the father node, the weight, and the active-settled status of the given node.

6 An example

We illustrate the algorithm on our 9-player peer group situation.

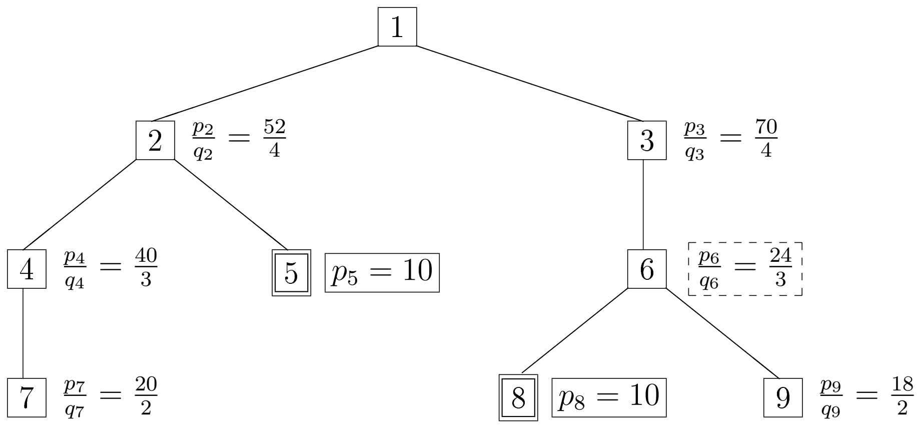
The algorithm starts with $\alpha = 0$ and $A = \{2, \ldots, 9\}$. After the initialization the p-value of a node shows the total weight of the subtree stemming from that node. After step *(L0)* of the first round the q-value gives the number of coalitions in the partition associated with that node.


1
2 $\frac{p_2}{q_2} = \frac{102}{5}$
3 $\frac{p_3}{q_3} = \frac{120}{5}$
4 $\frac{p_4}{q_4} = \frac{70}{3}$
5 $\frac{p_5}{q_5} = \frac{20}{2}$
6 $\frac{p_6}{q_6} = \frac{64}{4}$
7 $\frac{p_7}{q_7} = \frac{40}{2}$
8 $\frac{p_8}{q_8} = \frac{20}{2}$
9 $\frac{p_9}{q_9} = \frac{38}{2}$

Iteration 1 ($\alpha = 0$)

In this iteration the minimum ratio $\beta = 10$ is attained at nodes $B = \{5, 8\}$ (indicated by dashed boxes). Since both of them are leaves, B remains unchanged in step *(L2)*. Thus, nodes 5 and 8 become settled (indicated by double-lined boxes) with the increased value of $\alpha = 10$.

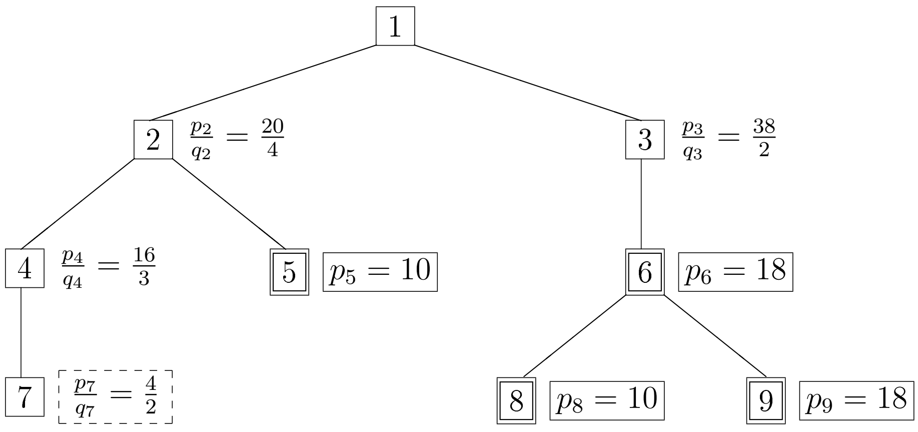
Iteration 2 starts with the updated tree with active nodes $A = \{2, 3, 4, 6, 7, 9\}$.


1
2 $\frac{p_2}{q_2} = \frac{52}{4}$
3 $\frac{p_3}{q_3} = \frac{70}{4}$
4 $\frac{p_4}{q_4} = \frac{40}{3}$
5 $p_5 = 10$
6 $\frac{p_6}{q_6} = \frac{24}{3}$
7 $\frac{p_7}{q_7} = \frac{20}{2}$
8 $p_8 = 10$
9 $\frac{p_9}{q_9} = \frac{18}{2}$

Iteration 2 ($\alpha = 10$)

In step *(L1)* we find $\beta = 8$ and $B = \{6\}$. Since node 9 is an active node in the subtree stemming from node 6, it also gets settled with the increased $\alpha = 18$, so after step *(L2)* we have $B = \{6, 9\}$.

In iteration 3 only $A = \{2, 3, 4, 7\}$ are the active nodes.

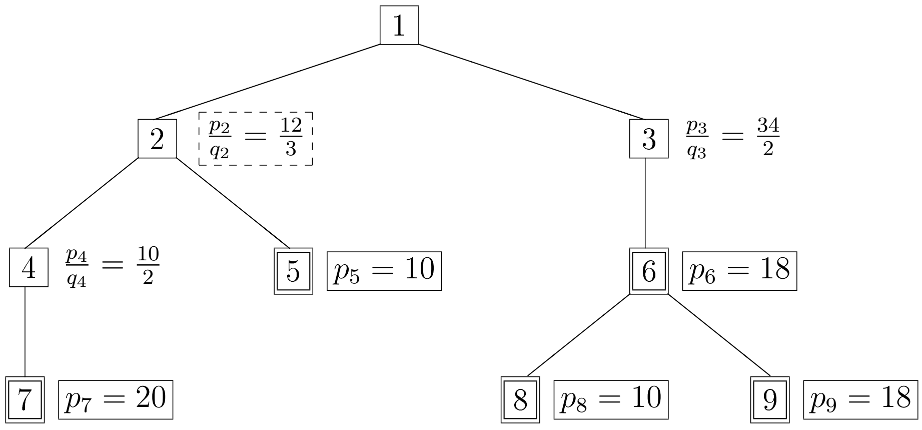


1
2
$\frac{p_2}{q_2} = \frac{20}{4}$
3
$\frac{p_3}{q_3} = \frac{38}{2}$
4
$\frac{p_4}{q_4} = \frac{16}{3}$
5
$p_5 = 10$
6
$p_6 = 18$
7
$\frac{p_7}{q_7} = \frac{4}{2}$
8
$p_8 = 10$
9
$p_9 = 18$

Iteration 3 ($\alpha = 18$)

In this round we get $\beta = 2$ and $B = \{7\}$.

Iteration 4 starts with active nodes $A = \{2, 3, 4\}$.

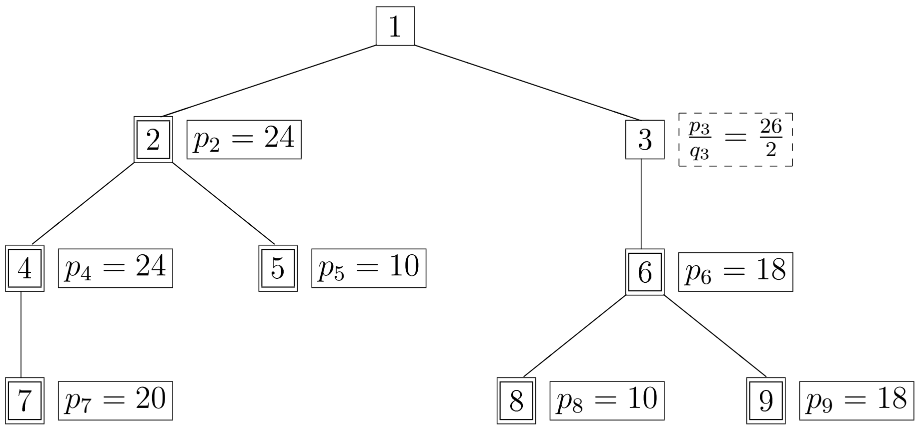


1
2
$\frac{p_2}{q_2} = \frac{12}{3}$
3
$\frac{p_3}{q_3} = \frac{34}{2}$
4
$\frac{p_4}{q_4} = \frac{10}{2}$
5
$p_5 = 10$
6
$p_6 = 18$
7
$p_7 = 20$
8
$p_8 = 10$
9
$p_9 = 18$

Iteration 4 ($\alpha = 20$)

In step *(L1)* we find $\beta = 4$ and $B = \{2\}$. Node 4 is also an active node in the subtree stemming from node 2, both gets settled with the increased $\alpha = 24$, so after step *(L2)* we have $B = \{2, 4\}$.

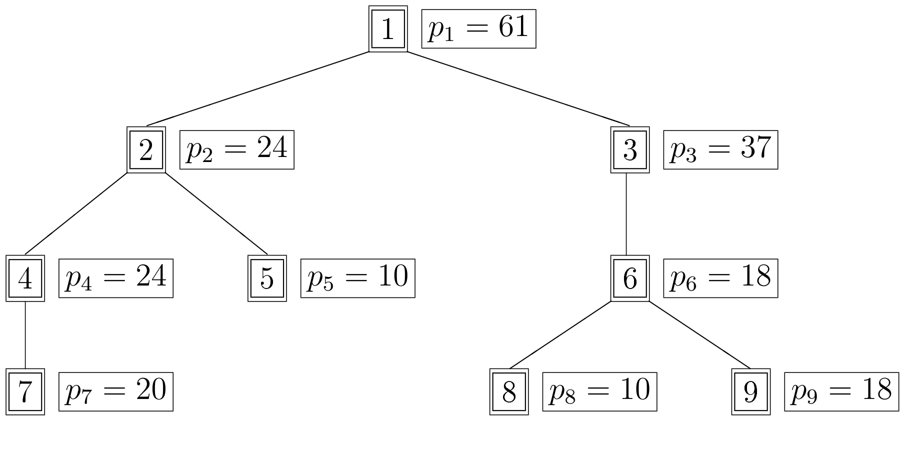
For iteration 5 we are left with only one active node $A = \{3\}$.



Iteration 5 ($\alpha = 24$)

In this round $\beta = 13$ and $B = \{3\}$, so the iterative phase ends after the updating steps.

Finally, after computing p_1 from efficiency, the algorithm stops with the nucleolus allocation values in variables p.



The nucleolus allocation

References

[1] Arin J, Feltkamp V (1997) The nucleolus and kernel of veto-rich transferable utility games. *International Journal of Game Theory,* 26: 61-74.

[2] Bondareva ON, Driessen TSH (1994) Extensive coverings and exact core bounds. *Games and Economic Behavior,* 6: 212-219.

[3] Brânzei R, Fragnelli V, Tijs S (2000) On the computation of the nucleolus of line-graph peer group games. *Scientific Annals of the "Alexandru Ioan Cuza" University,* Computer Science Section, tom IX, pp. 79-91.

[4] Brânzei R, Fragnelli V, Tijs S (2002) Tree-connected peer group situations and peer group games. *Mathematical Methods of Operations Research*, 55: 93-106.

[5] Brune S (1983) On the regions of linearity for the nucleolus and their computation. *International Journal of Game Theory*, 12: 47-80.

[6] Davis M, Maschler M (1965) The kernel of a cooperative game. *Naval Research Logistics Quarterly,* 12: 223-259.

[7] Granot D, Granot F, Zhu WR (1998) Characterization sets for the nucleolus. *International Journal of Game Theory,* 27: 359-374.

[8] Huberman G (1980) The nucleolus and the essential coalitions. In: Bensoussan A, Lions J (eds.) *Analysis and Optimization of Systems*, Lecture Notes in Control and Information Sciences 28, Springer, Berlin, pp. 416-422.

[9] Kuipers J, Solymosi T, Aarts H (2000) Computing the nucleolus of some combinatorially-structured games. *Mathematical Programming*, 88: 541-563.

[10] Le Breton M, Owen G, Weber S (1992) Strongly balanced cooperative games. *International Journal of Game Theory,* 20: 419-427.

[11] Maschler M, Peleg B, Shapley LS (1979) Geometric properties of the kernel, nucleolus and related solution concepts. *Mathematics of Operations Research*, 4: 303-338.

[12] Maschler M, Potters JAM, Tijs SH (1992) The general nucleolus and the reduced game property. *International Journal of Game Theory*, 21: 85-106.

[13] Potters JAM, Reijnierse J (1995) Γ-component additive games. *International Journal of Game Theory,* 24: 49-56.

[14] Reijnierse J, Potters JAM (1998) The $\mathcal{B}$-nucleolus of TU-games. *Games and Economic Behavior,* 24: 77-96.

[15] Schmeidler D (1969) The nucleolus of a characteristic function game. *SIAM Journal on Applied Mathematics*, 17: 1163-1170.

[16] Solymosi T, Raghavan TES, Tijs SH (2000) Computing the nucleolus of cyclic permutation games. *Manuscript.*