Mathematical Programming 83 (1998) 195–211

The nucleon of cooperative games and an algorithm for matching games

Ulrich Faigle [a,*], Walter Kern [a], Sándor P. Fekete [b], Winfried Hochstättler [b]

[a] *Department of Applied Mathematics, University of Twente, P.O. Box 217, NL-7500 AE Enschede, The Netherlands*

[b] *ZPR, Zentrum für paralleles Rechnen, Universität zu Köln, Albertus-Magnus-Platz, D-50923 Köln, Germany*

Received 9 October 1996; revised manuscript received 4 June 1997

Abstract

The *nucleon* is introduced as a new allocation concept for non-negative cooperative n-person transferable utility games. The nucleon may be viewed as the multiplicative analogue of Schmeidler's nucleolus. It is shown that the nucleon of (not necessarily bipartite) matching games can be computed in polynomial time. © 1998 The Mathematical Programming Society, Inc. Published by Elsevier Science B.V.

Keywords: Cooperative game; Core; Nucleolus; Nucleon; Matching

1. Introduction

One of the central problems in cooperative game theory is to provide fair allocations to the players in the game. The games that we consider here are cooperative n-person transferable utility games in characteristic function form. Formally, the general setup can be described as follows.

There is a finite set $N = \{1, \ldots, n\}$ of *players*. These players may form *coalitions* $S \subseteq N$ in an arbitrary way. Each coalition S can achieve a *value* $v(S) \in \mathbb{R}$ (assuming that the players in S "cooperate"). The value $v(N)$ of the *grand coalition* N can thus be understood as the total "profit" arising from the cooperation of all players. The pair (N, v) therefore represents our game in *characteristic form*. An *allocation* is a vector $x \in \mathbb{R}^N$ with component sum equal to $v(N)$. The allocations we seek should be *fair* in the sense that they assess the strength of individual players relative to (N, v) in an acceptable way.

* Corresponding author. E-mail. u.faigle@math.utwente.nl.

Published by Elsevier Science B.V.
PII S0025-5610(97)00102-0

Many interesting examples of such games have been investigated where the value $v(S)$ of a coalition $S \subseteq N$ is determined as the optimal value of a combinatorial optimization problem the set S of players faces (see, e.g., [1]).

In the *matching game*, for instance, we are given the complete graph K_n with N as the set of nodes. A *matching* is a set M of edges such that no two edges in M have a node in common. Each edge e in K_n is assigned a weight $w(e)$ and the value $v(S)$ of a coalition is equal to the weight of a maximal matching in the subgraph induced by S. Here each individual player $i \in N$ has value $v(i) = 0$ while value $v(N) > 0$ may well be possible. How should the strength of $i \in N$ be assessed?

There are many notions of "fairness" for allocations (see, e.g., [2]). In the following we will only present a few of them.

The idea of the *core* of a game, which essentially goes back to [3], approaches fairness from the point of view of coalitions. The allocation $x = (x_1, \dots, x_n)$ is said to be in the core of (N, v) if there is no coalition $S \subseteq N$ such that

$$\sum_{i \in S} x_i < v(S).$$

Note that the vectors x in the core of (N, v) form a polyhedron in $\mathbb{R}^N$ as they are determined by the linear restrictions:

$$\sum_{i \in N} x_i = v(N), \quad \sum_{i \in S} x_i \geqslant v(S) \quad \text{for all } S \subseteq N.$$

A game may have an empty core (e.g., the matching game on K_3 with unit edge weights). Therefore, relaxations of the concept of a core have received attention. For a given $\epsilon \in \mathbb{R}$, Shapley and Shubik [4] consider the modified game (N, v^ϵ), where

$$v^\epsilon(S) = \begin{cases} v(S) & \text{if } S \in \{\emptyset, N\}, \\ v(S) - \epsilon & \text{otherwise.} \end{cases}$$

The *additive ϵ-core* of the game (N, v) is defined to be the core of the game (N, v^ϵ).

Faigle and Kern [5] propose an ϵ-correction relative to the value of a proper coalition and arrive at the modified game (N, v_ϵ), where

$$v_\epsilon(S) = \begin{cases} v(S) & \text{if } S \in \{\emptyset, N\}, \\ (1 - \epsilon)v(S) & \text{otherwise.} \end{cases}$$

The *multiplicative ϵ-core* of (N, v) is then the core of the game (N, v_ϵ). The multiplicative ϵ-core is not the only meaningful way to relax the notion of the core. Indeed, several alternative "taxation models" have been introduced in the literature. For example, Shapley and Shubik [4] define a modified game by setting

$$v'_\epsilon(S) = \begin{cases} v(S) & \text{if } S \in \{\emptyset, N\}, \\ v(S) - \epsilon|S| & \text{otherwise,} \end{cases}$$

and Tijs and Driessen [6] propose the modification

$$v''_\epsilon(S) = \begin{cases} v(S) & \text{if } S \in \{\emptyset, N\}, \\ v(S) - \epsilon\left[v(S) - \sum_{i \in S} v(i)\right] & \text{otherwise.} \end{cases}$$

The multiplicative ϵ-core appears to be a natural concept in all situations where taxation is imposed proportionally to the value (e.g., sales tax). Note also that the multiplicative ϵ-core is equivalent with the taxation model of Tijs and Driessen [6] in the case of *zero-normalized* games, where $v(i) = 0$ holds for every individual player i. (The matching game studied in more detail below is of this kind.)

There is always some ϵ yielding a non-empty additive ϵ-core. The same is true for the multiplicative ϵ-core whenever $v(N) \geqslant 0$ (take, e.g., $\epsilon = 1$). For both models, this observation suggests to seek an ϵ that is as small as possible while still guaranteeing a non-empty ϵ-core (see, e.g., [5]) for the multiplicative ϵ-core of some combinatorial games.

Trying to rank the different allocation models according to some abstract qualitative merit is of little value. In the following example, the real world context and not a theoretical property determines the appropriateness of the allocation model.

Example. Consider a 2-person game with $v(1) = 1$, $v(2) = 2$, and $v(1,2) = 1000$. Computing the smallest ϵ with a non-empty multiplicative ϵ-core yields the unique allocation vector $x = (x_1, x_2) = (1000/3, 2000/3)$, while the additive version suggests the unique allocation $y = (y_1, y_2) = (999/2, 1001/2)$.

At first sight, one might consider allocation y to be more "reasonable" than x because their joint effort increases the return for both players by about the same amount. Suppose, on the other hand, that the two players' capital is 1 and 2 dollars, respectively and that they intend to buy a lottery ticket for 3 dollars to win the prize of 1000 dollars. The players must somehow agree in advance on how the 1000 dollars should be divided among the two investors. In this context now, allocation rule x, which rewards the players relative to their investments, seems quite appropriate. In fact, this way of allocating the total gain reflects economical behavior in many real life situations: think of two musicians producing a compact disc or the group of shareholders of a company. Here the *relative* sizes of the shares will determine the distribution of a potential gain among the partners.

The concept of the additive ϵ-core is refined by the notion of the *nucleolus* due to Schmeidler [8]. We want an allocation x that maximizes the *satisfaction*, i.e., the negative excess

$$e(x,S) = \sum_{i \in S} x_i - v(S)$$

uniformly over all proper coalitions S, i.e., we solve the linear program

$$(\mathrm{LP}_1) \qquad \max \ \epsilon$$
$$\sum_{i \in N} x_i = v(N),$$
$$\sum_{i \in S} x_i \geqslant v(S) + \epsilon \quad \text{for all } S \notin \{\emptyset, N\}.$$

Denoting by ϵ_1 the optimal objective function value of (LP_1), it follows that $\epsilon = -\epsilon_1$ is the minimal value admitting a non-empty additive ϵ-core.

The set of optimal solutions of (LP_1) is usually called the *prenucleolus* of the game. If (LP_1) has a unique solution (ϵ_1, x^*), then x^* is the *nucleolus* of the game (N, v). Otherwise, there is a unique collection $\mathscr{S}_1 \subset 2^N$ of coalitions S $(\neq \emptyset, N)$ for which the inequalities in (LP_1) become tight at $\epsilon = \epsilon_1$.

Now, in a second step, we maximize the satisfaction over all remaining coalitions:

$$
\begin{aligned}
(\mathrm{LP}_2) \qquad \max \quad & \epsilon \\
& \sum_{i \in N} x_i = v(N), \\
& \sum_{i \in S} x_i = v(S) + \epsilon_1 \quad \text{for all } S \in \mathscr{S}_1, \\
& \sum_{i \in S} x_i \geqslant v(S) + \epsilon \quad \text{otherwise.}
\end{aligned}
$$

Continuing in this way, we obtain a sequence

$$\epsilon_1 < \epsilon_2 < \cdots < \epsilon_k$$

until, finally, the optimal solution of (LP_k) is unique with an allocation x^*, the nucleolus of the game.

A more concise (and less algorithmic) description can be given as follows.

With the allocation x we associate the *satisfaction vector* $s(x) \in \mathbb{R}^{2^n-2}$ as the vector of negative excesses arranged in non-decreasing order. The nucleolus is then the unique vector x^* that lexicographically maximizes the satisfaction vectors $s(x)$ relative to the game (N, v).

General algorithms for the computation of the nucleolus have been investigated by several researchers (see, e.g., [7]). Relative to special classes of games, these algorithms do generally not guarantee a polynomially bounded running time. On the other hand, Solymosi and Raghavan [9] could show that the nucleolus of a matching game can be computed in polynomial time in the *bipartite* case, i.e., in the case where the edges of positive weight in the underlying graph do not contain a circuit of odd length.

We suggest another approach to the allocation problem for general matching games. In Section 2, we introduce the *nucleon* as the multiplicative analogue of the nucleolus for cooperative games in a straightforward way. From a purely mathematical point of view, the nucleon is a meaningful concept for general cooperative n-person games. From a conceptual point of view, however, there might be difficulties in accepting the multiplicative analogue of the excess of a coalition with *negative* value as an appropriate measure of its satisfaction. Therefore, we will restrict ourselves to games with non-negative characteristic functions.

Section 3 deals with complexity aspects of the nucleon in general terms. In Section 4, we focus on general matching games and, as an application of our new allocation concept, demonstrate that the nucleon of general matching games can be found in polynomial time.

We do not claim that the concept of the nucleon is always superior to the traditional nucleolus. There is probably something to be said in favor of any of the aforementioned ϵ-core taxation models and the same will be true for the corresponding variants of the idea of nucleolus. In view of the fact that the multiplicative ϵ-core has already attracted some interest (see, e.g., [10–12]), we expect that our multiplicative analogue of the nucleolus will prove to be useful in appropriate modeling contexts, too. There are many interesting questions to ask about the nucleon as a general solution concept. How does the nucleon behave with respect to scaling? When is the nucleon individually rational? How does the nucleon relate to other existing solution concepts? Since this paper is primarily concerned with computational issues, we leave an in-depth study of most of these questions open for future research. However, we would like to address here the following question: Given that the nucleon (cf. Section 2) is generally a set rather than a single vector, are there "natural candidates" allocations to choose? (Note that the same question, of course, applies to the core).

As a possible answer we propose to select the appropriate nucleon vector as to maximize a utility function that might be given on the set of players in particular instances. In the case where the nucleon allows a nice characterization (as it does with matching games), this approach is computationally feasible for linear utility functions. In the special case of matching games, this problem does not really occur since the nucleolus there is practically "always" a singleton (cf. Section 4).

Coming back to the motivating idea behind the present work, we want to emphasize the question of computational complexity in this context. There is no doubt that the usefulness of a concept depends both on its modeling adequacy *and* on its computational complexity. In the case of the general matching problem studied here, computation of the nucleon turns out to be polynomial while the computational complexity status of the nucleolus and any of its other variants is open.

2. The nucleon of a game

Let (N, v) be a cooperative n-person game. We will throughout assume that $v(\emptyset) = 0$ holds, i.e., that (N, v) is *normalized*. We will, furthermore, restrict our attention to non-negative games and thus assume that $v(S) \geqslant 0$ holds for any coalition $S \subseteq N$. To simplify the presentation, recall the (standard) notation relative to the vector $x \in \mathbb{R}^N$ and the coalition $S \subseteq N$

$$x(S) := \sum_{i \in S} x_i.$$

Let $\mathscr{S}_0 := \{\emptyset, N\}$ and $\alpha \geqslant 0$. Consider the polyhedron $P_1(\alpha)$ of all vectors x that satisfy the following linear restrictions:

$$\begin{aligned} P_1(\alpha) :: x(N) &= v(N), \\ x(S) &\geqslant \alpha v(S) \quad (S \notin \mathscr{S}_0). \end{aligned}$$

Letting $\alpha_0 := 0$, we conclude from the non-negativity of v

$$P_1(\alpha_0) = P_1(0) \neq \emptyset.$$

Moreover, $P_1(1)$ is precisely the (usual) core of the game (N, v).

Let

$$\alpha_1 := \max\{\alpha \in \mathbb{R} \mid P_1(\alpha) \neq \emptyset\}.$$

If $\alpha_1 = \infty$, we have $v(S) = 0$ for all $S \notin \mathscr{S}_0$. The nucleon $P^* = P^*(N, v)$ of the game (N, v) is then defined to be the polyhedron

$$P^* := P_1(\alpha_0) = \{x \in \mathbb{R}^N \mid x(N) = v(N), x \geqslant 0\}.$$

Otherwise, i.e., if $\alpha_1 < \infty$, let $\mathscr{S}_1$ denote the set of coalitions $S \subset N$ that correspond to "forced equalities" at level $\alpha = \alpha_1$, i.e.,

$$\mathscr{S}_1 := \{S \notin \mathscr{S}_0 \mid x(S) = \alpha_1 v(S) \text{ for all } x \in P_1(\alpha_1)\}.$$

Now assume that $P_j(\alpha), \alpha_j$, and $\mathscr{S}_j$ have been defined for $j = 1, \ldots, i$. Then let the polyhedron $P_{i+1}(\alpha)$ be defined by the linear constraints:

$$\begin{aligned} P_{i+1}(\alpha) :: x(N) &= v(N), \\ x(S) &= \alpha_1 v(S) \quad (S \in \mathscr{S}_1), \\ &\vdots \\ x(S) &= \alpha_i v(S) \quad (S \in \mathscr{S}_i), \\ x(S) &\geqslant \alpha v(S) \quad (S \notin \mathscr{S}_0 \cup \cdots \cup \mathscr{S}_i) \end{aligned}$$

and set

$$\alpha_{i+1} := \max\{\alpha \in \mathbb{R} \mid P_{i+1}(\alpha) \neq \emptyset\}.$$

If $\alpha_{i+1} = \infty$, then the nucleon of (N, v) is defined to be

$$P^* := P_i(\alpha_i) = P_{i+1}(\alpha_i).$$

Otherwise, i.e., if $\alpha_{i+1} < \infty$, set

$$\mathscr{S}_{i+1} := \{S \notin \mathscr{S}_0 \cup \cdots \cup \mathscr{S}_i \mid x(S) = \alpha_{i+1} v(S) \text{ for all } x \in P_{i+1}(\alpha_{i+1})\}$$

and continue.

Apparently, this inductive procedure will stop after a finite number of steps with $\alpha_{k+1} = \infty$ as soon as

$$v(S) = 0 \quad \text{for all } S \notin \mathscr{S}_0 \cup \cdots \cup \mathscr{S}_k.$$

Summarizing, the nucleon is obtained by successively computing

$$0 = \alpha_0 < \alpha_1 < \alpha_2 < \cdots < \alpha_k < \alpha_{k+1} = \infty,$$

$$P_0(0) \supseteq P_1(\alpha_1) \supseteq P_2(\alpha_2) \supseteq \cdots \supseteq P_k(\alpha_k) = P^*.$$

Example. Let $N = \{1, 2\}$, $v(N) = 1$, and $v(S) = 0$ otherwise. (This is the simplest case of a matching game on the complete graph K_2 with unit edge weight.) Then

$$P^* = \{(x_1, x_2) \in \mathbb{R}^2 \mid x_1 + x_2 = 1,\ x_1, x_2 \geqslant 0\}.$$

This example shows that the nucleon does not necessarily consist of a single vector $x^* \in \mathbb{R}^N_+$. However, if $\{i\} \subseteq \mathscr{S}_0 \cup \cdots \cup \mathscr{S}_k$ holds for all $i \in N$, then the nucleon P^* is a singleton. Indeed, in the latter case, the value $x_i = x(\{i\})$ is then prescribed at a fixed value for every $x \in P^*$. In particular, P^* will have cardinality one if $v(\{i\}) > 0$ for every $i \in N$.

As is the case for the (additive) nucleolus, there is an alternative definition of the nucleon in terms of "relative satisfaction". Given a game (N, v) and a vector $x \in \mathbb{R}^N_+$ with $x(N) = v(N)$, define for every coalition $S \notin \mathscr{S}_0$ the *satisfaction ratio* via

$$\alpha(x, S) := \begin{cases} x(S)/v(S) & \text{if } v(S) > 0, \\ \infty & \text{if } v(S) = 0. \end{cases}$$

The *relative satisfaction vector* $\alpha(x)$ is obtained by ordering the $2^n - 2$ satisfaction values $\alpha(x, S)$ in a non-decreasing sequence.

Proposition 2.1. *The nucleon of the non-negative game* (N, v) *is the set of all allocation vectors* $x \in \mathbb{R}^N_+$ *that lexicographically maximize the satisfaction vector* $\alpha(x)$.

We omit the straightforward proof of the proposition. A direct consequence is the following proposition.

Proposition 2.2. (*i*) *If the value* $v(N)$ *of the grand coalition is increased to* $\hat{v}(N) = cv(N)$ *for some* $c > 0$ (*and all other values remain the same*), *then the new nucleon equals* c *times the original one.*

(*ii*) *If a game* v *is modified to* $\hat{v} = cv$ *for some* $c > 0$, *then the new nucleon equals* c *times the original one.*

Proof. The map $x \to cx$ is a bijection between allocations for the game (N, v) and the modified game $(N, \hat{v})$. The mapping leaves the relative ordering of the satisfaction ratios invariant, which implies the proposition. □

The first part of Proposition 2.2 immediately implies that the nucleon is monotonic in the following sense: If $v(N)$ increases to $\hat{v}(N)$ (with all other values $v(S)$ unchanged), then for each x in the nucleon of v there exists an $\hat{x}$ in the nucleon of $\hat{v}$ such that $\hat{x} \geqslant x$. Increasing the value of a proper subcoalition, however, need not result in an increase of all individual allocations: if $v(1)$ is raised to 2 in the example of Section 1, player 2 will receive less than before. The nucleon relates to the multiplicative ϵ-core in the same way as the nucleolus to the additive ϵ-core. In particular, the nucleon and the nucleolus both lie in the core if the latter is non-empty. Because the core of an additive game is a singleton, we observe the following property.

Proposition 2.3. *The nucleon of a non-negative additive game equals the nucleolus.*

As a final remark on the relation between nucleon and nucleolus, recall that the nucleolus is covariant with strategic equivalence of games:

Let $v: 2^N \to \mathbb{R}$ be the characteristic function of a game, $\beta: 2^N \to \mathbb{R}$ an additive game, and $c \in \mathbb{R}$ a scalar.

If x is the nucleolus of the game (N, v) then the game $(N, cv + \beta)$ has nucleolus $cx + \beta'$, where β' is the restriction of β to the singletons. This property is certainly not shared by the nucleon unless $\beta = 0$ (Proposition 2.2 (ii)). In fact, the general idea of "strategic equivalence" contradicts in some sense the basic motivation behind the nucleon that the gain should be distributed according to the relative values of the subcoalitions. Part (i) of Proposition 2.2, which generally is false for the nucleolus, marks the distinction.

Note that our original "algorithmic" definition of the nucleon P^* does not provide an efficient way of computing a vector in P^*. Indeed, the sheer computation of α_1 in the way suggested by the definition means to solve a linear program with an exponential (in n) number of constraints. The question, therefore, arises whether P^* can be efficiently determined at all for interesting classes of games. We give a positive answer to this question for the special class of matching games in Section 4.

3. Computational aspects

Recall that the nucleon $P^* = P_k(\alpha_k)$ consists of all vectors x that satisfy the linear restrictions:

$$\begin{aligned} P_k(\alpha_k) :: x(N) &= v(N), \\ x(S) &= \alpha_1 v(S) \quad (S \in \mathscr{S}_1), \\ &\vdots \\ x(S) &= \alpha_k v(S) \quad (S \in \mathscr{S}_k), \\ x &\geqslant 0. \end{aligned}$$

The number k in the preceding definition of the nucleon P^* may, in general, be exponential in n. Intuitively, this can happen when all "new" equations

$$x(S) = \alpha_i v(S) \quad (S \in \mathscr{S}_i)$$

are already implied by the previous equations for $S \in \mathscr{S}_0 \cup \cdots \cup \mathscr{S}_{i-1}$. Then $\alpha_i > \alpha_{i-1}$ while $\dim P_i(\alpha_i) = \dim P_{i-1}(\alpha_{i-1})$. We want to derive an iterative computational procedure for P^* that avoids steps that are redundant in that sense. We will show that P^* can be found in at most n iterations.

For any $\mathscr{S} \subseteq 2^N$, denote by $\langle \mathscr{S} \rangle$ the *span* of $\mathscr{S}$, i.e.,

$$\langle \mathscr{S} \rangle := \{T \subsetneq N \mid \mathbb{1}_T \in \operatorname{lin}(\mathbb{1}_S \mid S \in \mathscr{S})\},$$

where $\mathbb{1}_S$ denotes the $(0,1)$-incidence vector of $S \subseteq N$ and *lin*(.) denotes the *linear hull* operator, i.e., for $X \subseteq \mathbb{R}^n$,

$$\mathrm{lin}(X) = \left\{ \sum_{i=1}^{n} \lambda_i x_i \mid \lambda_i \in \mathbb{R},\ x_i \in X \right\}.$$

With this terminology, we may describe P^* equivalently via

$$\begin{aligned} P^* :: \; & x(N) = v(N), \\ & x(S) = \alpha_1 v(S) \quad (S \in \mathscr{S}_1 \setminus \langle \mathscr{S}_0 \rangle), \\ & \quad \vdots \\ & x(S) = \alpha_k v(S) \quad (S \in \mathscr{S}_k \setminus \langle \mathscr{S}_0 \cup \cdots \cup \mathscr{S}_{k-1} \rangle), \\ & x \geqslant 0. \end{aligned}$$

This representation of P^* suggests the following iterative computational procedure: Let $\mathscr{T}_0 := \{\emptyset, N\}$ and define for $\beta \geqslant 0$ the polyhedron $Q_1(\beta)$ via

$$\begin{aligned} Q_1(\beta) \;\; :: \;\; & x(N) = v(N), \\ & x(T) \geqslant \beta v(T) \quad (T \notin \mathscr{T}_0). \end{aligned}$$

Note that for $\beta = 1 - \epsilon$, $Q_1(\beta)$ equals the multiplicative ϵ-core. Now set

$$\beta_1 \;:=\; \max\{\beta \in \mathbb{R} \mid Q_1(\beta) \neq \emptyset\}.$$

If $\beta_1 = \infty$, then

$$P^* \;=\; \{x \in \mathbb{R}^N_+ \mid x(N) = v(N)\}.$$

If $\beta_1 < \infty$, let $\mathscr{T}_1$ denote the set of coalitions that correspond to forced equalities at level $\beta = \beta_1$ (thus $\mathscr{T}_1 = \mathscr{S}_1$).

Now, assume inductively, that $Q_j(\beta), \beta_j$, and $\mathscr{T}_j$ have been defined for $j = 1, \ldots, i$. Let then the polyhedron $Q_{i+1}(\beta)$ be presented by the constraints:

$$\begin{aligned} Q_{i+1}(\beta) :: \; & x(N) = v(N), \\ & x(T) = \beta_1 v(T) \quad (T \in \mathscr{T}_1), \\ & \quad \vdots \\ & x(T) = \beta_i v(T) \quad (T \in \mathscr{T}_i), \\ & x(T) \geqslant \beta v(T) \;\; (T \notin \langle \mathscr{T}_0 \cup \cdots \cup \mathscr{T}_i \rangle), \end{aligned}$$

and set

$$\beta_{i+1} := \max\{\beta \in \mathbb{R} \mid Q_{i+1}(\beta) \neq \emptyset\}.$$

If $\beta_{i+1} = \infty$, then $P^* = Q_i(\beta_i)$ and we stop. Otherwise, define $\mathscr{T}_{i+1}$ to be the set of coalitions T that become tight at level $\beta = \beta_{i+1}$.

$$\mathscr{T}_{i+1} := \{T \notin \langle \mathscr{T}_0 \cup \cdots \cup \mathscr{T}_i \rangle \mid x(T) = \beta_{i+1} v(T) \text{ for all } x \in Q_{i+1}(\beta_{i+1})\}.$$

From the alternative description of P^* above, it is apparent that the sequence $(Q_i(\beta_i))$ is a subsequence of $(P_i(\alpha_i))$ and that (β_i) is a subsequence of (α_i).

There is another interpretation of the collections $\langle \mathscr{T}_0 \cup \cdots \cup \mathscr{T}_i \rangle$ occurring in the description of $Q_i(\beta_i)$ via linear inequalities. This interpretation will be particularly useful in the algorithmic analysis of the nucleon in Section 4.

Let $Q \subseteq \mathbb{R}^N$ be a set of vectors. We say that Q *fixes* the set $S \subseteq N$ if $x(S) = y(S)$ holds for all $x, y \in Q$. For $i = 1, \ldots, l$, let us define

$$\mathscr{F}_i := \{S \subseteq N \mid S \text{ is fixed by } Q_i(\beta_i)\}.$$

Lemma 3.1. *For* $i = 1, \ldots, l, \quad \mathscr{F}_i = \langle \mathscr{T}_0 \cup \cdots \cup \mathscr{T}_i \rangle$.

Proof. By definition, we have

$$\begin{aligned} Q_i(\beta_i) :: \ & x(N) = v(N), \\ & x(T) = \beta_1 v(T) \quad (T \in \mathscr{T}_1), \\ & \qquad \vdots \\ & x(T) = \beta_i v(T) \quad (T \in \mathscr{T}_i), \\ & x(T) \geqslant \beta_i v(T) \ \ (T \notin \langle \mathscr{T}_0 \cup \cdots \cup \mathscr{T}_i \rangle). \end{aligned}$$

Moreover, each of the inequalities for $x(T)$, $T \notin \langle \mathscr{T}_0 \cup \cdots \cup \mathscr{T}_i \rangle$, can be made strict. Hence, by taking convex combinations, we see that all these inequalities can be made strict simultaneously. Thus the relative interior $Q_i^o(\beta_i)$ is described by the constraints

$$\begin{aligned} Q_i^o(\beta_i) :: \ & x(N) = v(N), \\ & x(T) = \beta_1 v(T) \quad (T \in \mathscr{T}_1), \\ & \qquad \vdots \\ & x(T) = \beta_i v(T) \quad (T \in \mathscr{T}_i), \\ & x(T) > \beta_i v(T) \ \ (T \notin \langle \mathscr{T}_0 \cup \cdots \cup \mathscr{T}_i \rangle). \end{aligned}$$

It is now clear that the set of coalitions fixed by the relative interior $Q_i^o(\beta_i)$ is precisely $\langle \mathscr{T}_0 \cup \cdots \cup \mathscr{T}_i \rangle$. Hence $\mathscr{F}_i \subseteq \langle \mathscr{T}_0 \cup \cdots \cup \mathscr{T}_i \rangle$. The converse containment is straightforward. □

Returning to the complexity of computing the nucleon, note that in each iterative step in the computation of the sequence $(Q_i(\beta_i))$ equality constraints are added that are independent from the previous equality constraints. Hence we conclude for the dimension

$$\dim Q_{i+1}(\beta_{i+1}) \ < \ \dim Q_i(\beta_i),$$

which implies that P^* is determined after at most n iterations.

In each iteration, the parameter β_{i+1} is the optimal solution value of a linear program with the $n + 1$ variables $\beta, x_1, \ldots, x_n$. Hence the nucleon P^* can be determined by solving n linear programs successively. This direct procedure, however, will generally be not efficient because of the exponential (in n) size of the linear programs involved. Because we are interested in efficient algorithms for the computation of the nucleon P^*, we first want to demonstrate that the parameters β_i do not grow "too big" in the course of the iterative procedure.

Recall that the *size* $\langle\langle r\rangle\rangle$ of a rational number r is defined to be the number of bits in a binary representation of r. Then we observe the following proposition.

Proposition 3.1. *Let $\beta_1 < \cdots < \beta_l$ and $\mathscr{T}_1, \ldots, \mathscr{T}_l$ be given as above ($l \leqslant n$), and let $\mathscr{T} := \mathscr{T}_0 \cup \cdots \cup \mathscr{T}_l$. Then the size $\langle\langle\beta_i\rangle\rangle$ of each β_i is bounded by a polynomial in $n, |\mathscr{T}|$, and $\max_{T\in\mathscr{T}}\langle\langle v(T)\rangle\rangle$.*

Proof. It follows directly from the definition that $(\beta_1, \ldots, \beta_l)$ is the unique lexicographically maximal vector $(b_1, \ldots, b_l)$ such that the linear system

$$\begin{aligned} &x(N) = v(N), \\ &x(T) = b_1 v(T) \quad (T \in \mathscr{T}_1), \\ &\qquad\vdots \\ &x(T) = b_l v(T) \quad (T \in \mathscr{T}_l), \\ &x \geqslant 0 \end{aligned}$$

has a solution $x \in \mathbb{R}^N$. Hence we can obtain $(\beta_1, \ldots, \beta_l)$ from the unique lexicographically maximal solution $(b_1^*, \ldots, b_l^*, x_1^*, \ldots, x_n^*)$ of the above system.

The latter, however, represents a vertex of the feasibility region. Standard results from linear programming, therefore, imply that each component is polynomially bounded in the size of the system (see, e.g., [13]).

The size of the linear system is bounded by

$$\mathscr{O}((n + \max_{T\in\mathscr{T}}\langle\langle v(T)\rangle\rangle) \cdot |\mathscr{T}|) ,$$

which proves the proposition. □

4. The nucleon of a matching game

A matching game is defined on the graph $\mathscr{G} = (N, E)$ with an edge weighting $w : E \to \mathbb{R}$. The characteristic function v is given for each coalition $S \subseteq N$ via

$$v(S) = \text{value of a maximal weighted matching in } \mathscr{G}|_S ,$$

where $\mathscr{G}|_S$ is the subgraph of $\mathscr{G}$ induced by S.

Since a matching of maximal weight will never contain a negative edge we may assume w.l.o.g. that the weighting w is non-negative. Adding edges with weight zero, if necessary, we can similarly assume that $\mathscr{G}$ is the complete graph K_n.

Recall from Section 3 the inductively defined polyhedra $Q_i(\beta)$ and the interpretation offered by Lemma 3.1:

$$
\begin{aligned}
Q_i(\beta) :: x(N) &= v(N),\\
x(T) &= \beta_1 v(T) \quad (T \in \mathscr{T}_1),\\
&\vdots\\
x(T) &= \beta_{i-1} v(T) \quad (T \in \mathscr{T}_{i-1}),\\
x(T) &\geqslant \beta v(T) \quad (T \notin \langle \mathscr{T}_0 \cup \cdots \cup \mathscr{T}_{i-1} \rangle).
\end{aligned}
$$

Our aim is to show that the defining equations and inequalities for $Q_i(\beta)$ can be replaced by a polynomial number of equations and inequalities if we want to compute the nucleon of a matching game. Essentially, it will turn out that we may restrict our attention to the value an allocation x takes on one- and two-element coalitions. Unfortunately, however, this does *not* mean that the defining inequalities for $Q_i(\beta)$ are obtained by simply retaining the constraints corresponding to the one- and two-element coalitions. In order to obtain a polynomial algorithm, we have to proceed in a more subtle way.

While Lemma 3.1 is valid for arbitrary games, we will from now on assume that v arises from the matching game on $\mathscr{G}$ relative to the edge weighting w. We denote by $\mathscr{N}_i$ resp. $\mathscr{E}_i$ the one-resp, two-element coalitions in $\mathscr{F}_i$. We will usually think of $\mathscr{N}_i$ as a subset of N and of $\mathscr{E}_i$ as a subset of E.

Proposition 4.1. *For* $i = 1, \ldots, l$, $\mathscr{T}_i \subseteq \langle \mathscr{T}_0 \cup \mathscr{E}_i \cup \mathscr{N}_i \rangle$.

Proof. Consider any $S \in \mathscr{T}_i$. Let M be a matching of maximal weight in $\mathscr{G}|_S$, i.e., $v(S) = w(M)$. Because $\mathscr{G}$ is a complete graph and w is non-negative, we can also assume that M is a maximum cardinality matching, i.e., $S = N(M)$ if $|S|$ is even and $S = \{t\} \cup N(M)$ for some $t \in N$ if $|S|$ is odd. (For any set A of edges, we denote by $N(A)$ the nodes of $\mathscr{G}$ covered by A.) We will distinguish two cases.

Case 1: $M \subseteq \mathscr{E}_i$. If $|S|$ is even, then $S \in \langle \mathscr{E}_i \rangle \subseteq \langle \mathscr{T}_0 \cup \mathscr{E}_i \cup \mathscr{N}_i \rangle$. If $|S|$ is odd, then $S \in \mathscr{F}_i$ and $M \subseteq \mathscr{F}_i$ imply $\{t\} = S \setminus N(M) \in \mathscr{F}_i$. So $t \in \mathscr{N}_i$ and, therefore, $S = t \cup N(M) \in \langle \mathscr{T}_0 \cup \mathscr{E}_i \cup \mathscr{N}_i \rangle$.

Case 2: There exists some $e \in M \setminus \mathscr{E}_i$. Consider $S' := S \setminus N(e)$. If $S' \in \mathscr{F}_i$, then $S \in \mathscr{T}_i \subseteq \mathscr{F}_i$ implies that also $N(e) \in \mathscr{F}_i$ must hold, contrary to our assumption on e. So $S' \notin \mathscr{F}_i$ and, in particular, $S' \notin \langle \mathscr{T}_0 \cup \cdots \cup \mathscr{T}_{i-1} \rangle$.

By the definition of $Q_i(\beta_i)$, we know for all $x \in Q_i(\beta_i)$,

$$x(S') \geqslant \beta_i v(S').$$

On the other hand, we have $e \notin \langle \mathscr{T}_0 \cup \cdots \cup \mathscr{T}_{i-1} \rangle$ and, therefore, for all $x \in Q_i(\beta_i)$,

$$x(e) \geqslant \beta_i w(e).$$

Since $S \in \mathscr{T}_i$, we furthermore know for all $x \in Q_i(\beta_i)$, $x(S) = \beta_i v(S)$.

Summarizing, we conclude for all $x \in Q_i(\beta_i)$, $x(e) = \beta_i w(e)$, i.e., $e \in \mathscr{F}_i$, contrary to the choice of e. □

Proposition 4.1 calls our attention to the sequences

$$\begin{array}{lllllll} \emptyset = \mathscr{E}_0 & \subseteq & \mathscr{E}_1 & \subseteq & \cdots & \subseteq & \mathscr{E}_l, \\ \emptyset = \mathscr{N}_0 & \subseteq & \mathscr{N}_1 & \subseteq & \cdots & \subseteq & \mathscr{N}_l. \end{array}$$

In each iterative step $(i \to i+1)$ some edges $e \in \mathscr{E}_{i+1} \setminus \mathscr{E}_i$ become fixed by $Q_{i+1}(\beta_{i+1})$ to some non-negative value $c(e)$, say, until eventually all edges with non-zero weight are fixed.

Similarly, some nodes $t \in \mathscr{N}_{i+1} \setminus \mathscr{N}_i$ are fixed at some value $c(t) \geqslant 0$. The nucleon P^* is determined by

$$\begin{aligned} P^* :: \; & x(N) = v(N), \\ & x(e) = c(e) \quad (e \in \mathscr{E}_l), \\ & x(t) = c(t) \quad (t \in \mathscr{N}_l), \\ & x \geqslant 0. \end{aligned}$$

Furthermore, as a consequence of Proposition 4.1, we can describe $Q_i(\beta)$ via

$$\begin{aligned} Q_i(\beta) :: \; & x(N) = v(N), \\ & x(e) = c(e) \quad (e \in \mathscr{E}_{i-1}), \\ & x(t) = c(t) \quad (t \in \mathscr{N}_{i-1}), \\ & x(T) \geqslant \beta v(T) \quad (T \notin \langle \mathscr{T}_0 \cup \mathscr{E}_{i-1} \cup \mathscr{N}_{i-1} \rangle). \end{aligned}$$

Our next goal is to replace the exponentially many inequalities in the preceding description of $Q_i(\beta)$ by polynomially many inequalities (cf. the description of $Q_i^*(\beta)$ below).

For $i \geqslant 1$ and $\beta \geqslant 0$, let $\mathscr{G}_{i-1} = (N, \mathscr{E}_{i-1})$ be the subgraph containing only those edges that are fixed after the iterative step $i-1$. For $e \in E$, let $\mathscr{G}_{i-1} \setminus e$ denote the graph obtained from $\mathscr{G}_{i-1}$ by removing the two endpoints of e and all incident edges. Similarly, for $t \in N$, let $\mathscr{G}_{i-1} \setminus t$ be the subgraph obtained from $\mathscr{G}_{i-1}$ by removing t and all incident edges.

Relative to the original weighting $w\colon E \to \mathbb{R}$ and the (known) weighting $c\colon \mathscr{E}_{i-1} \to \mathbb{R}_+$, we define a new weighting $w_\beta\colon \mathscr{E}_{i-1} \to \mathbb{R}$ on $\mathscr{G}_{i-1}$ by

$$w_\beta(f) := \beta w(f) - c(f).$$

For $e \in E$, let M_β^e denote some fixed (possibly empty) matching in $\mathscr{G}_{i-1} \setminus e$ of maximal weight with respect to the weighting w_β. Let $S_\beta^e := N(e) \cup N(M_\beta^e)$ be the associated coalition. Similarly for $t \in N$, denote by M_β^t some fixed matching of maximal weight with respect to w_β in the graph $\mathscr{G}_{i-1} \setminus t$ and let $S_\beta^t := \{t\} \cup N(M_\beta^t)$ be the associated coalition.

Define the polyhedron $Q_i^*(\beta)$ by

$$\begin{aligned} Q^*(\beta) :: \; & x(N) = v(N), \\ & x(e) = c(e) \quad (e \in \mathscr{E}_{i-1}), \\ & x(t) = c(t) \quad (t \in \mathscr{N}_{i-1}), \\ & x(S_\beta^e) \geqslant \beta v(S_\beta^e) \quad (e \notin \mathscr{E}_{i-1}), \\ & x(S_\beta^t) \geqslant \beta v(S_\beta^t) \quad (t \notin \mathscr{N}_{i-1}). \end{aligned}$$

Proposition 4.2. $Q_i(\beta) = Q_i^*(\beta)$ *for all* $\beta \geqslant 0$.

Proof. By the choice of M_β^e, we have $N(M_\beta^e) \in \langle \mathscr{E}_{i-1} \rangle$. So $e \notin \mathscr{E}_{i-1}$, i.e, $e \notin \mathscr{F}_{i-1}$, implies $S_\beta^e \notin \mathscr{F}_{i-1}$. In particular, $e \notin \mathscr{E}_{i-1}$ yields $S_\beta^e \notin \langle \mathscr{T}_0 \cup \mathscr{E}_{i-1} \cup \mathscr{N}_{i-1} \rangle$. Therefore, all the inequalities occurring in the definition of $Q_i^*(\beta)$ also occur in the description of $Q_i(\beta)$. A similar argument holds for $t \notin \mathscr{N}_{i-1}$. Thus

$$Q_i(\beta) \subseteq Q_i^*(\beta) \quad \text{for all } \beta \geqslant 0.$$

Conversely, let $x \in Q_i^*(\beta)$ be arbitrary and let $S \notin \langle \mathscr{T}_0 \cup \mathscr{E}_{i-1} \cup \mathscr{N}_{i-1} \rangle$. We show that $x(S) \geqslant \beta v(S)$ holds, which implies $x \in Q_i(\beta)$.

Let M be a matching of maximum weight relative to w in $\mathscr{G}|_S$. So $v(S) = w(M)$. Assume again that M is of maximal cardinality, i.e., $S = N(M)$ or $S = t \cup N(M)$, depending on whether $|S|$ is even or odd.

Case 1: $M \subseteq \mathscr{E}_{i-1}$. Then $S = t \cup N(M)$ for some $t \notin \mathscr{N}_{i-1}$ (otherwise, we would have $S \in \langle \mathscr{E}_{i-1} \cup \mathscr{N}_{i-1} \rangle$, a contradiction to our assumption on S). Since $x \in Q_i^*(\beta)$, we know that

$$x(S_\beta^t) \geqslant \beta v(S_\beta^t).$$

Hence

$$\begin{aligned} x(t) &\geqslant \beta v(S_\beta^t) - x(M_\beta^t) \\ &\geqslant \beta w(M_\beta^t) - x(M_\beta^t) \\ &= w_\beta(M_\beta^t) \\ &\geqslant w_\beta(M) \\ &= \beta w(M) - x(M) \\ &= \beta v(S) - x(M). \end{aligned}$$

Thus

$$x(S) = x(t) + x(M) \geqslant \beta v(S).$$

Case 2: There exists some $e \in M \setminus \mathscr{E}_{i-1}$. First observe that $x \geqslant 0$ holds for all $x \in Q_i^*(\beta)$. Thus it suffices to show that $x(M) \geqslant \beta w(M)$ holds.

Let $U := M \cap \mathscr{E}_{i-1}$ and $V := M \setminus U$. For each $e \in V$, we have $x(S_\beta^e) \geqslant \beta v(S_\beta^e)$. So

$$\begin{aligned} x(e) &\geqslant \beta v(S_\beta^e) - x(M_\beta^e) \\ &\geqslant \beta w(e) + \beta w(M_\beta^e) - x(M_\beta^e) \\ &= \beta w(e) + w_\beta(M_\beta^e). \end{aligned}$$

Summing up the above inequalities for all $e \in V$ and using $w_\beta(M_\beta^e) \geqslant 0$ and $w_\beta(M_\beta^e) \geqslant w_\beta(U)$, we see

$$x(V) \geqslant \beta w(V) + w_\beta(U) = \beta w(V) + \beta w(U) - x(U).$$

Thus

$$x(M) \geqslant \beta(w(U) + w(V)) = \beta v(M). \qquad \square$$

We have achieved our goal. Given $\beta \geqslant 0$, we are able to represent $Q_i(\beta) = Q_i^*(\beta)$ with polynomially (in n) many equations and inequalities. Note that computing the coalitions S_β^t and S_β^e amounts to solving maximum weight matching problems with respect to the weighting w_β, which can be done in polynomial time (see, e.g., [14]), provided the weights w_β have polynomial size. One difficulty, however, remains. The coalitions of type S_β^e and S_β^t in the description of $Q_i^*(\beta)$ very much depend on the value of β. The idea, therefore, is to compute

$$\beta_i = \max\{\beta \mid Q_i^*(\beta) \neq \emptyset\}$$

by binary search.

Lemma 4.1. *For $i = 1, \ldots, l$,*

$$1/n \leqslant \beta_i \leqslant v(N)/w_{\min} =: M,$$

where $w_{\min}$ is the smallest non-zero weight $w(e)$, $e \in E$.

Proof. By definition,

$$\begin{aligned} Q_1^*(\beta) :: x(N) &= v(N), \\ x(e) &\geqslant \beta w(e) (e \in E). \end{aligned}$$

The vector $x = (v(N)/n, \ldots, v(N)/n)$ shows that $Q_1^*(1/n)$ is non-empty. Hence $1/n \leqslant \beta_1 < \cdots < \beta_l$. (In fact, one can show that $2/3 \leqslant \beta_1$ holds (see [5])).

On the other hand, each $\mathscr{T}_i$ contains at least some coalition T_i, say, with $v(T_i) > 0$ (otherwise $\beta_i = \infty$). But then $x(T_i) = \beta_i v(T_i) \geqslant \beta_i w_{\min}$ implies

$$\beta_i \leqslant x(T_i)/w_{\min} \leqslant x(N)/w_{\min} = M. \qquad \square$$

Now we are in the position to state our main result.

Theorem 4.1. *The nucleon P^* of a matching game on the graph $\mathscr{G} = (N, E)$ with edge weighting w can be computed in time polynomial in $n = |N|$ and the size $\langle\langle w \rangle\rangle$ of w.*

Proof. Let $s := \max\{\langle\langle w(e) \rangle\rangle \mid e \in E\}$. We know from Proposition 3.1 that each $\langle\langle \beta_i \rangle\rangle$ is polynomially bounded, say $\langle\langle \beta_i \rangle\rangle \leqslant p(n, s)$, for some suitable polynomial p.

It remains to deal with the size of $c : (\mathscr{E}_l \cup \mathscr{N}_l) \to \mathbb{R}$.

If $e \in \mathscr{E}_i \setminus \mathscr{E}_{i-1}$, then $x(e) = c(e)$ is determined by the equations

$$x(S_{\beta_i}^e) = \beta_i v(S_{\beta_i}^e)$$

and the (known) values that x takes on $\mathscr{E}_0 \cup \cdots \cup \mathscr{E}_{i-1}$ and $\mathscr{N}_0 \cup \cdots \cup \mathscr{N}_{i-1}$. Therefore, the nucleon P^* may alternatively be described via

$$
\begin{aligned}
P^* :: x(N) &= v(N) \\
x(S^e_{\beta_1}) &= \beta_1 v(S^e_{\beta_1}) \quad (e \in \mathscr{E}_1), \\
x(S^t_{\beta_1}) &= \beta_1 v(S^t_{\beta_1}) \quad (e \in \mathscr{N}_1), \\
&\vdots \\
x(S^e_{\beta_l}) &= \beta_l v(S^e_{\beta_l}) \quad (e \in \mathscr{E}_l), \\
x(S^t_{\beta_l}) &= \beta_l v(S^t_{\beta_l}) \quad (e \in \mathscr{N}_l), \\
x &\geqslant 0.
\end{aligned}
$$

This system has size polynomial in n and s. Consequently, any basic solution x of that system has size polynomial in n and s. But each basic solution x fixes $c(e) = x(e)$ and $c(t) = x(t)$ for every $e \in \mathscr{E}_l$ and $t \in \mathscr{N}_l$. Therefore, the size $\langle\langle c \rangle\rangle$ of c is polynomial.

The latter fact ensures that we can successively compute $\beta_1, \ldots, \beta_l$ in time polynomial in n and s by applying binary search to determine β_i in each iterative step. □

How large can the nucleon of a matching game be? Obviously, if $\mathscr{G}$ consists of exactly one edge $e = (1,2)$ with weight $w(e) > 0$, the nucleon is the line segment

$$P^* = \{x \in \mathbb{R}^2_+ \mid x_1 + x_2 = w(e)\}.$$

However, this is about the only case where the nucleon is not a singleton.

Theorem 4.2. *If $\mathscr{G} = (N, E)$ contains at least two edges with positive weight, then the nucleon P^* of the associated matching game is a singleton.*

Proof. Observe that the nucleon P^* fixes each coalition S with $v(S) > 0$. If $e = (s,t) \in E$ has positive weight $w(e) > 0$ and $u \in N$ is not an endpoint of e, then P^* fixes both $\{s,t\}$ and $\{s,t,u\}$. Hence P^* fixes each $u \notin \{s,t\}$. Thus, if $\mathscr{G}$ has two distinct edges e_1 and e_2 of positive weight, it follows that P^* fixes every node $u \in N$. □

Calculating the nucleon is quite a lengthy (though polynomial) affair even for non-trivial matching games with a relatively small number n of players as the algorithm requires the solution of at least n linear programs. Therefore, we refrain from presenting an explicit numerical example here. Finding a practically efficient algorithm for calculating the nucleon is still an open problem. Note, however, that Faigle and Kern [5] exhibit a general bound $\alpha_1 \geqslant 1 - 1/k$, where k is the length of the smallest odd cycle with positive edge weights. This implies that the nucleon will allocate to every coalition at least 66% of its value (and, if the graph is triangle-free, even at least 80%).

Acknowledgements

We thank the referee for his careful reading and insightful comments on the article.

References

[1] I. Curiel, Cooperative Game Theory and Applications, Kluwer, Dordrecht, 1997.
[2] M. Shubik, Game theory models and methods in political economy, In: K.J. Arrow et al. (Eds.), Handbook of Mathematical Economics, vol. 1, North-Holland, New York, 1981.
[3] J. von Neumann, O. Morgenstern, Theory of Games and Economic Behavior, Princeton Univ. Press, Princeton, NJ, 1944.
[4] L.S. Shapley, M. Shubik, Quasi-cores in a monetary economy with nonconvex preferences, Econometrica 34 (1966) 805–827.
[5] U. Faigle, W. Kern, On some approximately balanced combinatorial cooperative games, ZOR – Methods and Models of Operations Research 38 (1993) 141–152.
[6] S.H. Tijs, T.S.H. Driessen, Extensions of solution concepts by means of multiplicative ϵ-tax games, Mathematical Social Sciences 12 (1986) 9–20.
[7] J.A.M. Potters, J.H. Reijnierse, M. Ansing, Computing the nucleolus by solving a prolonged simplex algorithm, Mathematics of Operation Research 21 (1996) 757–768.
[8] D. Schmeidler, The nucleolus of a characteristic function game, SIAM Journal of Applied Mathematics 17 (1969) 1163–1170.
[9] T. Solymosi, T.E.S. Raghavan, An algorithm for finding the nucleolus of assignment games, International Journal of Game Theory 23 (1994) 119–143.
[10] U. Faigle, S.P. Fekete, W. Hochstättler, W. Kern, 1995. On approximately fair cost allocation in Euclidean TSP games, OR Spektrum (to appear).
[11] G.J. Woeginger, On the rate of taxation in a cooperative bin packing game, Methods and Models of Operations Research 42 (1995) 313–324.
[12] J. Kuipers, Oral communication (1995).
[13] M. Grötschel, L. Lovász, A. Schrijver, Geometric Algorithms and Combinatorial Optimization, Springer, Heidelberg, 1988.
[14] L. Lovász, M.D. Plummer. Matching theory, in: North-Holland Mathematical Studies, vol. 121, North-Holland, Amsterdam, 1986.