

Neighbor games and the leximax solution

Citation for published version (APA):

Klijn, F., Vermeulen, A. J., Hamers, H., Solymosi, T., Nakabeppu, Y., & Villar, J. P. (2003). Neighbor games and the leximax solution. *Mathematical Methods of Operations Research*, 58, 191-208.
<https://doi.org/10.1007/s001860300298>

Document status and date:

Published: 01/01/2003

DOI:

[10.1007/s001860300298](https://doi.org/10.1007/s001860300298)

Document Version:

Publisher's PDF, also known as Version of record

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.umlib.nl/taverne-license

Take down policy

If you believe that this document breaches copyright please contact us at:

repository@maastrichtuniversity.nl

providing details and we will investigate your claim.

Neighbor games and the leximax solution

Flip Klijn^{a, 1}, Dries Vermeulen^b, Herbert Hamers^c, Tamás Solymosi^{d, 2},
Stef Tijs^c, Joan Pere Villar^a

^a Dept. d'Economia i d'Història Econòmica, Universitat Autònoma de Barcelona, Spain

^b Department of Economics, University of Maastricht, The Netherlands

^c Department of Econometrics and CentER, Tilburg University, The Netherlands

^d Department of Operations Research, Budapest University of Economic Sciences, Hungary

¹ CODE and Departament d'Economia i d'Història Econòmica,

Universitat Autònoma de Barcelona, Edifici B, 08193 Bellaterra (Barcelona),

Spain (Tel.: (34) 93 581 1720; Fax: (34) 93 581 2012; e-mail: fklijn@pareto.uab.es)

Manuscript received: September 2002/Final version received: March 2003

Abstract. Neighbor games arise from certain matching or sequencing situations in which only some specific pairs of players can obtain a positive gain. As a consequence, the class of neighbor games is the intersection of the class of assignment games (Shapley and Shubik (1972)) and the class of component additive games (Curiel *et al.* (1994)). We first present some elementary features of neighbor games. After that we provide a polynomially bounded algorithm of order p^3 for calculating the leximax solution (cf. Arin and Iñarra (1997)) of neighbor games, where p is the number of players.

Key words: Neighbor Games, Leximax Solution, Assignment Games

1 Introduction

In this paper we introduce neighbor games and provide an algorithm to calculate the leximax solution (cf. Arin and Iñarra (1997)) of neighbor games. The following two examples describe situations that give rise to neighbor games.

In the first example we consider a sequencing situation in which customers are lined up in a queue and waiting for a taxi. The taxi company that provides the service has two types of cars: one that transports only one customer (type A) and one that can only transport two customers (type B). The first customer in the queue can decide to pick a taxi of type A or wait for the next customer in the queue. In the latter case they decide both to share a taxi of type B or the second customer will wait for the third customer. In the latter case the first

² This author's work has been supported by CentER and the Department of Econometrics, Tilburg University and by the Foundation for the Hungarian Higher Education and Research (AMFK).

customer has to pick a taxi of type A. This procedure is repeated until all customers are transported in a taxi. Since the costs of sharing a taxi of type B are lower than taking two taxis of type A, it is obvious that the customers can save costs by sharing a taxi of type B. However, each customer faces the problem that the cost of a taxi (of type B) is not fixed, because it depends on the trip to bring the customers to the right locations. Hence, we have that only customers that are neighbors in the queue can obtain cost savings, and customers that take a taxi of type A have cost savings equal to zero. All customers in the queue want to choose a combination of taxis of type A and B such that their cost savings are maximized. Moreover, they are looking for an allocation of the cost savings that is 'stable'.

The second example can be viewed as a restricted matching problem. Suppose a river runs through a number of regions. To be able to utilize this cheap transportation possibility, harbours have to be built. Because of financial restrictions, each country is able to build at most one harbour. Neighbor regions might join to build a harbour at their border (which then can serve both regions) and save costs. The regions are interested in maximizing their cost savings and finding some proper allocation of the cost savings.

For analyzing both examples we can use cooperative game theory, since one of the topics in cooperative game theory is the investigation of the stability of allocation rules. For this purpose we introduce neighbor games. In neighbor games, players are lined up in a one-dimensional queue. In this queue, players can only directly cooperate with one of their neighbors in the queue.

It turns out that the class of neighbor games is the intersection of the class of assignment games (Shapley and Shubik (1972)) and the class of component additive game (cf. Curiel et al. (1994)). The latter one is a the class of Γ -component additive games (cf. Potters and Reijnierse (1995)) in which the restricted graph is a line graph. As a consequence, neighbor games have many appealing properties, such as: the core is a non-empty set and coincides with the set of competitive equilibria (Shapley and Shubik (1972)), the core coincides with the bargaining set, and the nucleolus coincides with the kernel (Potters and Reijnierse (1995)). Moreover, neighbor games satisfy the CoMa-property, i.e., the core is the convex hull of some marginal vectors (cf. Hamers et al. (2002)).

In this paper we study in detail the leximax solution (cf. Arin and Iñarra (1997)) for neighbor games. The leximax solution is an egalitarian solution that equals the core allocation that minimizes the maximum satisfaction among all players. Note that there is some relation with the nucleolus (cf. Schmeidler (1969)), since the nucleolus maximizes the minimum satisfaction among all non-empty coalitions of players. The nucleolus for neighbor games is studied in Hamers et al. (2003).

The leximax solution and its natural counterpart the leximin solution are investigated for several classes of games. In Arin and Iñarra (1997) the leximin solution is studied for the class of convex games and veto games that are monotonic with respect to the grand coalition. Arin et al. (1998) studied the leximax solution on the class of large core games. Since the class of neighbor games is not a subclass of any of the above mentioned classes of games we study the leximax solution for neighbor games. We characterize the leximax solution in terms of adjustability to egalitarianism, which induces an

algorithm for finding the leximax solution. This algorithm is shown to be of order p^3 . A nice feature of the algorithm is that it can be visualized nicely by pictures, showing the process of adjusting and fixing the payoffs of the players.

In Section 2 we introduce neighbor games, relate them with other classes of games, and provide a convexity result. In Section 3 we characterize the leximax solution for the class of neighbor games. The proof of this characterization will be used in Section 4 to provide an $\mathcal{O}(p^3)$ algorithm for finding the leximax solution.

2 Neighbor games

In this section we introduce the class of neighbor games and present some results on the core of neighbor games. But we start with recalling some notions of cooperative game theory. In particular, we recall the definition of two classes of games that are very closely related to neighbor games: assignment games and component additive games.

A *cooperative game with transferable utility* (or game, for short) is a pair (P, v) where $P = \{1, \dots, p\}$ is a finite set of players and $v : 2^P \rightarrow \mathbb{R}$ is a map that assigns to each coalition $S \in 2^P$ a real number $v(S)$, such that $v(\emptyset) = 0$. Here, 2^P is the collection of all subsets (coalitions) of P .

The core of a game (P, v) consists of all vectors that distribute the gains $v(P)$ obtained by P among the players in such a way that no subset of players can be better off by seceding from the rest of the players and act on their own behalf. Formally, the *core* of a game (P, v) is defined by

$$\text{Core}(P, v) := \{x \in \mathbb{R}^P : x(S) \geq v(S) \text{ for all } S \subset P \text{ and } x(P) = v(P)\},$$

where $x(S) := \sum_{i \in S} x_i$ for $S \subseteq P$. A game (P, v) is called *balanced* if $\text{Core}(P, v) \neq \emptyset$.

A game (P, v) is called *convex* if for all $i \in P$ and all coalitions S and T with $S \subset T \subseteq P \setminus \{i\}$ it holds that

$$v(S \cup \{i\}) - v(S) \leq v(T \cup \{i\}) - v(T).$$

Assignment games, introduced by Shapley and Shubik (1972), arise from bipartite matching situations. Let M and N be two finite and disjoint sets. For each $i \in M$ and $j \in N$ the value of a matched pair (i, j) is $a_{ij} \geq 0$. From this situation an assignment game is defined in the following way. The worth of a coalition $S \cup T$ where $S \subseteq M$ and $T \subseteq N$ is defined to be the maximum that $S \cup T$ can achieve by making suitable pairs from its members. If $S = \emptyset$ or $T = \emptyset$ no suitable pairs can be made and therefore the worth in this situation is 0. Formally, an *assignment game* $(M \cup N, w)$ is defined by

$$w(S \cup T) := \max \left\{ \sum_{(i,j) \in \mu} a_{ij} : \mu \in \mathcal{M}(S, T) \right\} \quad \text{for all } S \subseteq M, T \subseteq N,$$

where $\mathcal{M}(S, T)$ denotes the set of matchings between S and T , i.e., collections of disjoint pairs (i, j) with $i \in S$ and $j \in T$.

The class of component additive games, introduced by Curiel et al. (1994), is a special class of Γ -component additive games, discussed in Potters and Reijnierse (1995), which in turn is a special class of graph restricted games in

the sense of Owen (1986). Let (P, v) be a cooperative game and let $\Gamma = (P, E)$ be an undirected line graph. Then a *component additive game* (P, w_Γ) is defined by

$$w_\Gamma(S) := \sum_{T \in S \setminus \Gamma} v(T) \quad \text{for all } S \subseteq P,$$

where $S \setminus \Gamma$ is the set of connected components of S with respect to Γ .

The situations discussed in the introduction that motivate the interest for neighbor games, give rise to a model in which players are lined up in a one-dimensional queue. In the queue, players can only directly cooperate with one of the neighbors in the queue. From this point of view, neighbor games are defined as restricted assignment games: only pairs that are neighbors in the queue can be matched. Formally, let P be the player set of size p . For the sake of convenience we assume that $P = \{1, \dots, p\}$. Without loss of generality we may assume that the players are ordered $1 \prec 2 \prec \dots \prec p$. Players i and j are called *neighbors* if $|i - j| = 1$. A *matching* μ for $Q \subseteq P$ is a (possibly empty) collection of disjoint pairs $(i, i + 1)$ of neighboring players (*partners*) in Q . Henceforth, the word matching means a matching of this type. Let $\mathcal{N}(Q)$ denote the set of matchings for Q . For all pairs of neighbors $(i, i + 1)$ let $a_{ii+1} \geq 0$ be given. Then, a *neighbor game* (P, v) is defined by

$$v(Q) := \max \left\{ \sum_{(i,j) \in \mu} a_{ij} : \mu \in \mathcal{N}(Q) \right\} \quad \text{for all } Q \subseteq P.$$

Note that since $a_{ii+1} = v(i, i + 1)$ a neighbor game is completely determined by the values of the pairs of neighbors. Note also that $v(i) = 0$ for all $i \in P$. A matching $\mu \in \mathcal{N}(Q)$ is called *optimal* for Q if $\sum_{(i,i+1) \in \mu} a_{ii+1} = v(Q)$. It is called *minimal* for Q if $a_{ii+1} > 0$ for all $(i, i + 1) \in \mu$. Throughout this section and with a slight abuse of notation, we identify a (possibly non-matched) pair $(i, i + 1)$ of neighbors in P with the two-person coalition $\{i, i + 1\}$. Let $Q \subseteq P$ and $\mu \in \mathcal{N}(Q)$. Let $i \in P$. If $(i - 1, i) \in \mu$ or $(i, i + 1) \in \mu$ then player i is called *matched* (with respect to μ), otherwise he is called *isolated* (with respect to μ).

Example 2.1. Let $P = \{1, 2, 3, 4\}$ be the player set. Take $a_{12} = 10, a_{23} = 20$, and $a_{34} = 30$. Then the corresponding neighbor game (P, v) is depicted in Table 1. The matching $\mu = \{(1, 2), (3, 4)\}$ is optimal and minimal for P . $\square$

The following proposition follows immediately from the definition of neighbor games. The proof is therefore omitted.

Proposition 2.2. *The class of neighbor games is the intersection of the class of assignment games and component additive games.*

Since neighbor games are special assignment games, the results of Shapley and Shubik (1972) on the core of assignment games apply to the core of

Table 1. A neighbor game (P, v)

S	$\{1, 2\}$	$\{2, 3\}$	$\{3, 4\}$	$\{1, 2, 3\}$	$\{1, 2, 4\}$	$\{1, 3, 4\}$	$\{2, 3, 4\}$	$\{1, 2, 3, 4\}$
$v(S)$	10	20	30	20	10	30	30	40

neighbor games. In particular, the core of neighbor games is not empty. Furthermore, it is determined by the inequalities induced by the one player coalitions and the pairs of neighbors. Henceforth, whenever we speak of a coalition it is a singleton or a pair of neighbors.

Let (N, v) be a neighbor game. Let μ be an optimal matching for P . With a slight abuse of notation we denote by P^+ the set of players that are matched by μ . The set of isolated players is denoted by $P^- = P \setminus P^+$. The following lemma is a straightforward consequence of a result of Shapley and Shubik (1972). Its proof is therefore omitted.

Lemma 2.3. *Let (P, v) be a neighbor game. Let μ be an optimal matching for P . Let $x \in \mathbb{R}^P$. Then, $x \in \text{Core}(P, v)$ if and only if the following four conditions are satisfied:*

- (i) $x_i + x_{i+1} = v(i, i+1)$ for all $(i, i+1) \in \mu$;
- (ii) $x_i + x_{i+1} \geq v(i, i+1)$ for all $(i, i+1) \notin \mu$;
- (iii) $x_i = 0$ for all players $i \in P^-$;
- (iv) $x_i \geq 0$ for all players $i \in P^+$.

In general, a neighbor game does not need to be convex, as follows from the next proposition, which provides a necessary and sufficient condition for the convexity of neighbor games.

Proposition 2.4. *A neighbor game (P, v) is convex if and only if for any triple $j-1, j, j+1 \in P$ of consecutive players it holds that $v(j-1, j) = 0$ or $v(j, j+1) = 0$.*

Proof. We first prove the ‘only if’-part. Suppose that $v(j-1, j) > 0$ and $v(j, j+1) > 0$ for some $j \in P$. Then,

$$\begin{aligned} v(j-1, j, j+1) - v(j-1, j) &= \max\{v(j-1, j), v(j, j+1)\} - v(j-1, j) \\ &= \max\{0, v(j, j+1) - v(j-1, j)\} \\ &< v(j, j+1) - v(j). \end{aligned}$$

Hence, (P, v) is not convex.

To prove the ‘if’-part, suppose that for any triple $j-1, j, j+1 \in P$ of consecutive players it holds that $v(j-1, j) = 0$ or $v(j, j+1) = 0$. Take $S \subset T \subset P$ and $k \in P \setminus T$. It is easy, but tedious, to check that

$$\begin{aligned} v(T \cup \{k\}) - v(T) &= \sum_{i \in A \cap T} v(i, k) \\ &\geq \sum_{i \in A \cap S} v(i, k) \\ &= v(S \cup \{k\}) - v(S), \end{aligned}$$

where A is the set defined by

$$A := \begin{cases} \{k-1, k+1\} & \text{if } k \neq 1, p; \\ \{2\} & \text{if } k = 1; \\ \{p-1\} & \text{if } k = p. \end{cases}$$

This proves the convexity of (P, v) . □

So, neighbor games are not convex in general. Hence, the core of a neighbor game does not need to be the convex hull of *all* marginal vectors. Nevertheless, since neighbor games are assignment games, it follows from Hamers et al. (2002) that they satisfy the CoMa-property, i.e., the core is the convex hull of *some* marginal vectors.

3 The leximax solution, a characterization

In this section we recall the leximax solution, a solution concept that was introduced by Arin and Iñarra (1997). After that, we characterize the restriction of the leximax solution to the class of neighbor games in terms of adjustability to egalitarianism.

Before we turn to the definition of the leximax solution, we first recall the notion of lexicographical ordering. Given two vectors $x, y \in \mathbb{R}^q$ for some q , we have that $x \preceq_{lex} y$ if either $x = y$ or there exists an index k such that $x_i = y_i$ for $i = 1, \dots, k$ and $x_{k+1} < y_{k+1}$. Further, let $\hat{x}$ be the vector that results when arranging the elements of the vector x in a non-increasing order, i.e., $\hat{x}_1 \geq \hat{x}_2 \geq \dots \geq \hat{x}_q$. Then, for a balanced game (P, v) , Arin and Iñarra (1997) defined the *leximax solution* $Lmax(P, v)$ as

$$Lmax(P, v) := \{x \in Core(P, v) : \hat{x} \preceq_{lex} \hat{y} \text{ for all } y \in Core(P, v)\}.$$

So, the leximax solution minimizes lexicographically the maximum payoff among all core allocations. Arin and Iñarra (1997) showed that the leximax solution is a one-point solution. This fact also follows from Lemma 1.1 of Moulin (1988) in which a leximax-like solution for bargaining situations is studied.

Theorem 3.1. *For a balanced game (P, v) , $Lmax(v)$ is a singleton.*

For a balanced game (P, v) we henceforth identify $Lmax(P, v)$ with its unique element.

Arin and Iñarra (1997) provided an algorithm that determines the leximin solution (the natural counterpart of the leximax solution) for convex games and veto games that are P -monotonic. Recall that from Proposition 2.4 it follows that in general neighbor games are not convex. A game (P, v) is called a veto game if there is a player $i \in P$ such that $v(S) = 0$ for all $S \subseteq P \setminus \{i\}$. A game (P, v) is called P -monotonic if $v(P) \geq v(S)$ for all $S \subset P$. It is clear from the definition of a neighbor game that in general neighbor games are not veto games.

The leximax solution was also studied by Arin et al. (1998). They provided a characterization of the leximax solution on the class of large core games, which are defined next. Let (P, v) be a balanced game. We define $U(P, v)$ as the set of games (P, w) with $w(S) = v(S)$ for all $S \neq P$ and $w(P) \geq v(P)$. Then, the game (P, v) is said to have a large core (Sharkey (1982)) if for all $(P, w) \in U(P, v)$ and for all $x \in Core(w)$ there exists an allocation $y \in Core(v)$ such that $y_i \leq x_i$ for all $i \in P$. The next example shows that in general neighbor games do not have a large core.

Example 3.2. Let (P, v) be the neighbor game with $P = \{1, 2, 3\}$ (in the order $1 \prec 2 \prec 3$) and $v(1, 2) = 6$ and $v(2, 3) = 10$. Then, as is easily verified,

$$Core(v) = \{\lambda(0, 6, 4) + (1 - \lambda)(0, 10, 0) : 0 \leq \lambda \leq 1\}.$$

Now, consider the game $(P, w) \in U(P, v)$ with $w(P) = 14$. Notice that $x = (4, 2, 8) \in \text{Core}(w)$, but there is no $y \in \text{Core}(v)$ such that $y_2 \leq x_2$ (since for all $y \in \text{Core}(v)$ we have $y_2 \geq 6 > 2 = x_2$). Hence, the neighbor game (P, v) does not have a large core. $\square$

From the above it follows that the known results and algorithms concerning the leximax solution cannot be applied to the class of neighbor games. Hence, for the determination of the leximax solution for neighbor games we need to develop a new algorithm. Before this we first provide a characterization of the leximax solution in terms of adjustability to egalitarianism.

Let (P, v) be a neighbor game. Let μ be an optimal matching. A pair $(i, i+1)$ is called *essential* if $(i, i+1) \in \mu$. A coalition $I \subseteq P$ is called an interval if $i, j \in I$ and $i \leq k \leq j$ imply that $k \in I$. We write $I = [i, j]$ for an interval $I \subseteq P$ if i and j are the starting point and the end point of I , respectively.

Definition 3.3. Let (P, v) be a neighbor game. Let $x \in \text{Core}(P, v)$ be a core allocation. An interval $[i-1, k]$ ($k \geq i$) is called *s-relevant*³ for player $i \in P$ with respect to x , if it satisfies the following three conditions:

- (1) $(i, i+1)$ is either not essential or non-existent (i.e., $i = p$);
- (2) x is tight on $[i-1, k]$ (i.e., $x_j + x_{j+1} = v(j, j+1)$ for all $j, j+1 \in [i-1, k]$);
- (3) $[i-1, k] \subseteq P^+$ (so essential and non-essential pairs alternate on $[i-1, k]$).

For intervals of the form $[k, i+1]$, p-relevancy is defined in a similar way.

Definition 3.4. Let (P, v) be a neighbor game. Let $x \in \text{Core}(P, v)$ be a core allocation. An interval $[k, i+1]$ ($k \leq i$) is called *p-relevant*⁴ for player $i \in P$ with respect to x , if it satisfies the following three conditions:

- (1) $(i-1, i)$ is either not essential or non-existent (i.e., $i = 1$);
- (2) x is tight on $[k, i+1]$;
- (3) $[k, i+1] \subseteq P^+$.

If an interval is s-relevant (p-relevant) for a player i with respect to a core allocation x , we say, when no confusion is possible, that the interval is s-relevant (p-relevant) for player i . An interval I is called *relevant* for player $i \in P$ if it is s-relevant or p-relevant for player i .

Lemma 3.5. Let (P, v) be a neighbor game. Let $x \in \text{Core}(P, v)$ be a core allocation.

- (i) If $i \in P^-$, then no interval is relevant for i .
- (ii) If $i \in P^+$, then i has either only s-relevant intervals or only p-relevant intervals.
- (iii) If $i \in P^+$, then i has a unique maximal relevant interval.

Proof. (i) follows from condition (3) of s-relevancy and p-relevancy.

(ii) Since $i \in P^+$ we have that either $(i-1, i)$ or $(i, i+1)$ is essential. Then condition (1) of s-relevancy and p-relevancy proves this part of the lemma.

(iii) is a straightforward consequence of statement (ii) of the lemma. $\square$

³ The s stands for successor.

⁴ The p stands for predecessor.

The *maximal relevant interval* for a player $i \in P^+$ with respect to a core allocation x is henceforth denoted by $I(i, x)$. From Lemma 3.5 it follows that $I(i, x) \neq \emptyset$ for all $i \in P^+$.

Lemma 3.6. *Let (P, v) be a neighbor game. Let $x \in \text{Core}(P, v)$. For $i \in P^+$, the cardinality of $I(i, x)$ is even.*

Proof. By Lemma 3.5 (ii) we have that $I(i, x) = [i - 1, k]$ or $I(i, x) = [k, i + 1]$ for some $k \in P$. We may assume, without loss of generality, that $I(i, x)$ is of the form $[i - 1, k]$. Then, by condition (3) of s-relevancy we have that $k \in P^+$. Then, $(k, k + 1)$ cannot be essential. Otherwise, $[i - 1, k + 1]$ would be s-relevant for i , which would contradict the maximality of $I(i, x)$. Hence, it follows readily, since essential and inessential pairs alternate, that $|I(i, x)|$ is even. $\square$

In the following definition we define adjustability of the payoff of a matched player. This notion will be used in the characterization of the leximax solution.

Definition 3.7. Let (P, v) be a neighbor game. Let $x \in \text{Core}(P, v)$. The payoff x_i of a player $i \in P^+$ can be *adjusted*⁵ with respect to x if the following three conditions are satisfied:

- (1) $x_j > 0$ for all $j \in I(i, x)$ with $|i - j|$ even;
- (2) $x_j < x_i$ for all $j \in I(i, x)$ with $|i - j|$ odd;
- (3) (a) If $I(i, x)$ is of the form $[i - 1, k]$, then either $k + 1$ is non-existent or $x_k + x_{k+1} > v(k, k + 1)$.
 (b) If $I(i, x)$ is of the form $[k, i + 1]$, then either $k - 1$ is non-existent or $x_{k-1} + x_k > v(k - 1, k)$.

Before we can characterize the leximax solution we need the following technical lemma.

Lemma 3.8. *Let $x, y \in \mathbb{R}^P$ with $\hat{x} \neq \hat{y}$ and $\hat{y} \preceq_{\text{lex}} \hat{x}$. Let $\sigma : \{1, \dots, p\} \rightarrow P$ be a bijection such that $x_{\sigma(1)} \geq x_{\sigma(2)} \geq \dots \geq x_{\sigma(p)}$. Let r be the smallest number with $x_{\sigma(r)} > y_{\sigma(r)}$. Then for all $l < r$, $x_{\sigma(l)} = y_{\sigma(l)}$.*

Proof. By induction on the number of players p . For $p = 1, 2$ the statement is quite obvious. Assume that the lemma holds for $p - 1$ for some $p \geq 3$. If $r = 1$, the lemma holds trivially. If $r > 1$, then distinguish between $l = 1$ and $2 \leq l < r$.

Case 1: $l = 1$. Since $x_{\sigma(1)}$ is the maximal coordinate of x , $x_{\sigma(1)} \leq y_{\sigma(1)}$ (since $r > 1$). So, since $\hat{y} \preceq_{\text{lex}} \hat{x}$, it is clear that $x_{\sigma(1)} = y_{\sigma(1)}$.

Case 2: $2 \leq l < r$. Consider the restrictions of x and y to $P \setminus \{\sigma(1)\}$ and apply the induction hypothesis. $\square$

⁵ For the sake of convenience we will say that a player itself can (or cannot) be adjusted with respect to x .

Theorem 3.9. *Let (P, v) be a neighbor game. Let $x \in \text{Core}(P, v)$. Then, $x = \text{Lmax}(P, v)$ if and only if no player $i \in P^+$ can be adjusted with respect to x .*

Proof. We first prove the ‘only if’-part. Suppose that some player $i \in P^+$ can be adjusted with respect to x . We will show that there is a core allocation $y \in \text{Core}(P, v)$ with $\hat{y} \neq \hat{x}$ and $\hat{y} \preceq_{\text{lex}} \hat{x}$. Assume, without loss of generality, that $I(i, x) = [i - 1, k]$ for some k . Since i can be adjusted, there exists $\epsilon > 0$ such that for all $j \in [i - 1, k]$

- (A1) $x_j - \epsilon > 0$ if $|i - j|$ is even;
- (A2) $x_j + \epsilon < x_i - \epsilon$ if $|i - j|$ is odd;
- (A3) $x_k + x_{k+1} - \epsilon > v(k, k + 1)$ if $k + 1 \in P$;
- (A4) $x_j < x_i - \epsilon$ for all $j \notin [i - 1, k]$ with $x_j < x_i$.

Now define $y \in \mathbb{R}^P$ by

$$y_j := \begin{cases} x_j & \text{if } j \notin I(i, x); \\ x_j + \epsilon & \text{if } j \in I(i, x) \text{ and } |i - j| \text{ odd}; \\ x_j - \epsilon & \text{if } j \in I(i, x) \text{ and } |i - j| \text{ even.} \end{cases} \quad (1)$$

Since $I(i, x) \neq \emptyset$, it follows that $y \neq x$.

We will prove that $y \in \text{Core}(P, v)$ by checking the conditions in Lemma 2.3.

(i) $(j, j + 1) \in \mu$.

Note that then either $j, j + 1 \in I(i, x)$ or $j, j + 1 \notin I(i, x)$. If $j, j + 1 \in I(i, x)$, then $y_j + y_{j+1} = (x_j \pm \epsilon) + (x_{j+1} \mp \epsilon) = x_j + x_{j+1} = v(j, j + 1)$. If $j, j + 1 \notin I(i, x)$, then $y_j + y_{j+1} = x_j + x_{j+1} = v(j, j + 1)$. So, in either case, $y_j + y_{j+1} = v(j, j + 1)$.

(ii) $(j, j + 1) \notin \mu$.

We distinguish among three cases.

Case a: $j, j + 1 \in I(i, x)$ or $j, j + 1 \notin I(i, x)$.

A proof similar to that of (i) shows that $y_j + y_{j+1} \geq v(j, j + 1)$.

Case b: $j \in I(i, x)$, $j + 1 \notin I(i, x)$.

Obviously, $j = k$. By Lemma 3.6 we have that $|i - j| = |i - k|$ is even. Hence, by the definition of y we have that $y_j = x_j - \epsilon$ and $y_{j+1} = x_{j+1}$. So, $y_j + y_{j+1} = x_j - \epsilon + x_{j+1} > v(j, j + 1)$, where the inequality follows from (A3).

Case c: $j \notin I(i, x)$, $j + 1 \in I(i, x)$.

Obviously, $j + 1 = i - 1$. So, $|i - (j + 1)| = |i - (i - 1)|$ is odd. Hence, by the definition of y we have that $y_{j+1} = x_{j+1} + \epsilon$ and $y_j = x_j$. So, $y_j + y_{j+1} = x_j + (x_{j+1} + \epsilon) \geq v(j, j + 1)$, where the inequality follows from $x \in \text{Core}(P, v)$.

(iii) $j \in P^-$.

Then, since $I(i, x) \subseteq P^+$, $j \notin I(i, x)$. So, $y_j = x_j = 0$.

(iv) $j \in P^+$.

If $j \in I(i, x)$, then by (A1) of the choice of ϵ and the definition of y , it follows that $y_j \geq 0$.

If $j \notin I(i, x)$, then $y_j = x_j \geq 0$.

Hence, $y \in \text{Core}(P, v)$.

Now, we will show that $\hat{y} \preceq_{\text{lex}} \hat{x}$ and $\hat{y} \neq \hat{x}$. Let $J := \{j \in P : y_j \neq x_j\} = I(i, x)$. Take $k \in \arg \max_{j \in J} x_j$. Then, $y_k \neq x_k$ and $x_k \geq x_j$ for all $j \in J$. Since $k \in J$ we have either $y_k = x_k - \epsilon$ or $y_k = x_k + \epsilon$. Suppose that $y_k = x_k + \epsilon$. Then, $y_k = x_k + \epsilon < x_i - \epsilon \leq x_k - \epsilon$. The first inequality follows from (A2) and the second inequality from $i \in J$ and the fact that $x_k \geq x_j$ for all $j \in J$. So, we have a contradiction. Hence, $y_k = x_k - \epsilon$.

Now take $l \in J$ with $y_l = x_l + \epsilon$. We have that $y_l = x_l + \epsilon < x_i - \epsilon \leq x_k - \epsilon = y_k$. Again, the first inequality follows from (A2) and the second inequality from $i \in J$ and the fact that $x_k \geq x_j$ for all $j \in J$. We conclude that $y_l < y_k$ for all $l \in J$ with $y_l = x_l + \epsilon$.

From $y_k = x_k - \epsilon$ and $y_l < y_k$ for all $l \in J$ with $y_l = x_l + \epsilon$ it follows that $\hat{y} \preceq_{\text{lex}} \hat{x}$ and $\hat{y} \neq \hat{x}$.

Now we will prove the ‘if’-part. Suppose there is a core allocation $y \in \text{Core}(P, v)$ with $\hat{y} \neq \hat{x}$ and $\hat{y} \preceq_{\text{lex}} \hat{x}$. Let $\sigma : \{1, \dots, p\} \rightarrow P$ be a bijection such that $x_{\sigma(1)} \geq x_{\sigma(2)} \geq \dots \geq x_{\sigma(p)}$. We may assume, without loss of generality, that σ satisfies the following condition: if $y_{\sigma(\alpha)} < x_{\sigma(\alpha)} = x_{\sigma(\beta)} \leq y_{\sigma(\beta)}$, then $\alpha > \beta$.

Let r be the smallest number with $x_{\sigma(r)} > y_{\sigma(r)}$. (Note that this r exists, because $x \neq y$.) We claim that player $\sigma(r)$ can be adjusted with respect to x . First notice that $\sigma(r) \in P^+$, since $x_{\sigma(r)} > y_{\sigma(r)} \geq 0$, where the second inequality follows from $y \in \text{Core}(P, v)$. Now we check conditions (I), (2), and (3) of Definition 3.7.

- (1) Take $j \in I(\sigma(r), x)$ for which $|j - \sigma(r)|$ is even. From $x_{\sigma(r)} > y_{\sigma(r)}$, $y \in \text{Core}(P, v)$, and condition (2) of Definition 3.3 and Definition 3.4 it follows that $x_j > y_j \geq 0$.
- (2) Take $j \in I(\sigma(r), x)$ for which $|j - \sigma(r)|$ is odd. Assume that $x_j \geq x_{\sigma(r)}$. From $x_{\sigma(r)} > y_{\sigma(r)}$, $y \in \text{Core}(P, v)$, and condition (2) of Definition 3.3 and Definition 3.4 it follows that $y_j > x_j$. By the assumption on σ and $x_j \geq x_{\sigma(r)}$ there is a number $l < r$ with $\sigma(l) = j$. This, however, contradicts Lemma 3.8. So, $x_j < x_{\sigma(r)}$.
- (3) We may assume, without loss of generality, that $I(\sigma(r), x) = [\sigma(r) - 1, m]$ for some $m \geq \sigma(r)$. Suppose that $m + 1$ exists. We prove that $x_m + x_{m+1} > v(m, m + 1)$. We distinguish between two cases.

Case 1: $m + 2$ does not exist or $(m + 1, m + 2)$ is not essential. In both cases, $m + 1 \in P^-$.

Then, by $x, y \in \text{Core}(P, v)$, $m + 1 \in P^-$, and Lemma 2.3 (iii), we have $x_{m+1} = 0 = y_{m+1}$.

Since $|m - \sigma(r)|$ is even by Lemma 3.6, we know that $x_m > y_m$ as in (I). Hence, $x_m + x_{m+1} > y_m + y_{m+1} \geq v(m, m + 1)$.

Case 2: $(m + 1, m + 2)$ is essential. Then, by definition of $I(\sigma(r), x)$, x is not tight on $\{m, m + 1\}$. So, $x_m + x_{m+1} > v(m, m + 1)$. $\square$

In the following proposition we provide a closed formula for the leximax solution of neighbor games in case there are four or less players involved.

Proposition 3.10. *Let (P, v) be a two person neighbor game, where $P = \{1, 2\}$ and the characteristic function v is induced by $a_{12} = a \geq 0$. Then $Lmax(P, v) = (\frac{a}{2}, \frac{a}{2})$.*

Let (P, v) be a three person neighbor game, where $P = \{1, 2, 3\}$ and the characteristic function v is induced by $a_{12} = a \geq 0$ and $a_{23} = b \geq 0$. Assume, without loss of generality, that $a \geq b$. Then⁶ $Lmax(P, v) = (\frac{a}{2} \wedge (a - b), \frac{a}{2} \vee b, 0)$.

Let (P, v) be a four person neighbor game, where $P = \{1, 2, 3, 4\}$ and the characteristic function v is induced by $a_{12} = a \geq 0$, $a_{23} = b \geq 0$, and $a_{34} = c \geq 0$. Assume, without loss of generality, that $a \geq c$. Then,

- (i) *if $b \in [0, \frac{a+c}{2}]$, then $Lmax(P, v) = (\frac{a}{2}, \frac{a}{2}, \frac{c}{2}, \frac{c}{2})$.*
- (ii) *if $b \in (\frac{a+c}{2}, \frac{a+2c}{2}]$, then $Lmax(P, v) = (\frac{a}{2} \wedge (a - \frac{b}{2}), \frac{a}{2} \vee \frac{b}{2}, (b - \frac{a}{2}) \wedge \frac{b}{2}, (c + \frac{a}{2} - b) \vee (c - \frac{b}{2}))$.*
- (iii) *if $b \in (\frac{a+2c}{2}, a + c)$, then $Lmax(P, v) = (0 \vee (c - \frac{b}{2}), c \wedge \frac{b}{2}, (b - c) \vee \frac{b}{2}, (a + c - b) \wedge (a - \frac{b}{2}))$.*
- (iv) *if $b \in [a + c, \infty)$, then $Lmax(P, v) = (0, \frac{b}{2} \vee a, \frac{b}{2} \wedge (b - a), 0)$.*

Proof. One easily checks the conditions in Definition 3.7 to see that no player is adjustable. Then the proposition follows from Theorem 3.9. $\square$

4 The leximax solution, an algorithm

In this section we provide an algorithm for finding the leximax solution for neighbor games. A nice feature of the algorithm is that it can be visualized nicely by some pictures showing the process of adjusting and fixing payoffs. We first present the algorithm. Then, we give an illustrative example. After that, we give a formal proof that the algorithm does indeed yield the leximax solution. Finally, we show that the algorithm is polynomially bounded of order p^3 in the number of players p .

Let (P, v) be a neighbor game and let μ be an optimal matching for P . The algorithm to find $Lmax(P, v)$ is based on the proof of Theorem 3.9. Loosely speaking, given an initial allocation, the algorithm generates a more egalitarian solution thereby fixing the payoffs of some players in P^+ . The algorithm terminates when the payoffs of all players in P^+ are fixed. The final allocation is the leximax solution, since – as we will see later – whenever we fix the payoff of a particular player, that player is no longer adjustable in the remainder of the algorithm.

⁶ For $x, y \in \mathbb{R}$ we define $x \wedge y := \min\{x, y\}$ and $x \vee y := \max\{x, y\}$.

Algorithm for the leximax solution for neighbor games
Input

A neighbor game (P, v) .

A core allocation⁷ $x \in \text{Core}(P, v)$.

Initialisation

Let μ be an optimal matching for P .

Let P^+ be the set of players that are matched by μ .

Set $F := \emptyset$. We call $F \subseteq P^+$ the set of *fixed players*.

Recursive step

Step 1. If $F = P^+$, then **STOP**, $L\max(P, v) = x$. Otherwise, define⁸

$$S^1 := \{i \in P^+ \setminus F : x_i \geq x_j \text{ for all } j \in P^+ \setminus F\}.$$

Step 2. Calculate the set C^1 of inadjustable players in S^1 .

If $C^1 \neq \emptyset$, say $C^1 = \{i_1, \dots, i_k\}$, then set $t := 1$ and do the following procedure:

Beginning of the F-procedure

If $t \leq k$, take $i := i_t$.

Otherwise, skip the procedure.

If $I(i, x) = [i - 1, k]$, then:

Step a. If there is a player $m \in I(i, x)$ with $x_m = 0$ and $|i - m|$ even, then set $F := F \cup [i - 1, m]$.

Step b. If there is a player $m \in I(i, x)$ with $x_m \geq x_i$ and $|i - m|$ odd, then take the player m^* with the highest index satisfying $m^* \in I(i, x)$ with $x_{m^*} \geq x_i$ and $|i - m^*|$ odd. Set $F := F \cup [i - 1, m^* + 1]$.

Step c. If player $k + 1$ exists and $x_k + x_{k+1} = v(k, k + 1)$ (so, $k + 1 \in P^-$), then set $F := F \cup [i - 1, k]$.

If $I(i, x) = [k, i + 1]$, then:

Step a. If there is a player $m \in I(i, x)$ with $x_m = 0$ and $|i - m|$ even, then set $F := F \cup [m, i + 1]$.

Step b. If there is a player $m \in I(i, x)$ with $x_m \geq x_i$ and $|i - m|$ odd, then take the player m^* with the lowest index satisfying $m^* \in I(i, x)$ with $x_{m^*} \geq x_i$ and $|i - m^*|$ odd. Set $F := F \cup [m^* - 1, i + 1]$.

Step c. If player $k - 1$ exists and $x_{k-1} + x_k = v(k - 1, k)$ (so, $k - 1 \in P^-$), then set $F := F \cup [k, i + 1]$.

Set $t := t + 1$ and repeat the procedure.

⁷ A core allocation can for example be obtained by solving a certain linear programming problem (cf. Shapley and Shubik (1972)). Another possibility is calculating the nucleolus using the algorithm in Hamers et al. (2003). This takes $\mathcal{O}(p^2)$ time.

⁸ Notice that the set S^1 is not empty.

End of the F-procedure

If $S^1 \subseteq F$, then **go to Step 1**.

If $S^1 \not\subseteq F$, then define

$$S^2 := S^1 \setminus F \neq \emptyset.$$

Step 3. For $\epsilon > 0$, consider the conditions (B1), (B2), (B3), and (B4) for a player $i \in S^2$.

(B1) $x_j - \epsilon > 0$ if $j \in I(i, x)$ and $|i - j|$ is even;

(B2) $x_j + \epsilon < x_i - \epsilon$ if $j \in I(i, x)$ and $|i - j|$ is odd;

(B3)(a) $x_k + x_{k+1} - \epsilon > v(k, k+1)$ if $I(i, x) = [i-1, k]$ and $k+1 \in P$;

(B3)(b) $x_{k-1} + x_k - \epsilon > v(k-1, k)$ if $I(i, x) = [k, i+1]$ and $k-1 \in P$;

(B4) $x_j < x_i - \epsilon$ for all $j \notin \bigcup_{l \in S^2} I(l, x)$ with $x_j < x_i$.

Beginning of the x-procedure

Calculate the smallest positive number $\epsilon > 0$ for which one of the conditions (B1), (B2), (B3), and (B4) becomes an equality for one of the players $i \in S^2$. Define the allocation $y \in \mathbb{R}^P$ by

$$y_j := \begin{cases} x_j & \text{if } j \notin \bigcup_{i \in S^2} I(i, x); \\ x_j + \epsilon & \text{if } j \in I(i, x), |i - j| \text{ odd, and } i \in S^2; \\ x_j - \epsilon & \text{if } j \in I(i, x), |i - j| \text{ even, and } i \in S^2. \end{cases} \quad (2)$$

Set $x := y$.

End of the x-procedure**Repeat recursive step**

In the following example we visualize the algorithm, showing the process of adjusting and fixing the payoffs of the players.

Example 4.1. Consider the neighbor game (P, v) where $P = \{1, \dots, 9\}$ is the set of players. Let v be the characteristic function determined by the values of the neighbors as given in Table 2. One readily verifies that there is a unique optimal matching, viz., the matching $\mu = \{(1, 2), (3, 4), (5, 6), (7, 8)\}$. As initial allocation we take $x = (0, 3, 7, 3, 0, 3, 1, 5, 0)$.

The game (P, v) and the allocation x are depicted in Figure 1. We put the players along the horizontal axis and their respective payoffs along the vertical axis. We connect the payoffs of the players so that the allocation x corresponds to a piecewise linear graph. Moreover, using Lemma 2.3 we immediately see that x is a core allocation:

Table 2. The values of the neighbors in the neighbor game (P, v)

S	$\{1, 2\}$	$\{2, 3\}$	$\{3, 4\}$	$\{4, 5\}$	$\{5, 6\}$	$\{6, 7\}$	$\{7, 8\}$	$\{8, 9\}$
$v(S)$	3	10	10	3	3	4	6	4

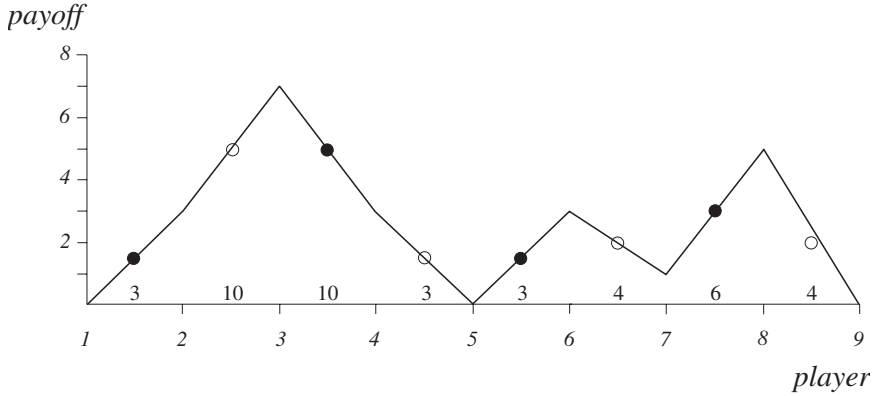


Fig. 1. The initial allocation x

- (i) The line through the payoffs of two matched neighbors runs exactly through the filled circle, which denotes half of the value of these neighbors;
- (ii) The line through the payoffs of two unmatched neighbors lies above or runs through the open circle, which denotes half of the value of these neighbors;
- (iii) All matched players receive a non-negative payoff;
- (iv) The unmatched player receives a payoff equal to zero.

We apply the algorithm to x to find the leximax solution for the game (P, v) . Note that $P^+ = \{1, \dots, 8\}$ and $P^- = \{9\}$. Set $F := \emptyset$.

Loop I: ($F \neq P^+$)

Step 1: $S^1 = \{3\}$.

Step 2: $C^1 = \{3\}$, since player 3 is not adjustable (Definition 3.7 (I) with $j = 1$). As a consequence, $F = \{1, 2, 3, 4\}$. Since $S^1 \subseteq F$ we go to Loop II.

Loop II: ($F = \{1, 2, 3, 4\} \neq P^+$)

Step 1: $S^1 = \{8\}$.

Step 2: $C^1 = \emptyset$. Hence, $S^2 = \{8\}$.

Step 3: $I(8, x) = [7, 8]$ and by condition (B3)(a) with $k + 1 = 9$ we have $\epsilon = 1$. The new allocation x is depicted in Figure 2.

Loop III: ($F = \{1, 2, 3, 4\} \neq P^+$)

Step 1: $S^1 = \{8\}$.

Step 2: $C^1 = \{8\}$, since player 8 is not adjustable (Definition 3.7 (3)(a) with $k + 1 = 9$). As a consequence, $F = \{1, 2, 3, 4, 7, 8\}$. Since $S^1 \subseteq F$ we go to Loop IV.

Loop IV: ($F = \{1, 2, 3, 4, 7, 8\} \neq P^+$)

Step 1: $S^1 = \{6\}$.

Step 2: $C^1 = \emptyset$. Hence, $S^2 = \{6\}$.

Step 3: $I(6, x) = [5, 6]$ and by condition (B3)(a) with $k + 1 = 7$ and condition (B4) with $j = 7$ we have $\epsilon = 1$. The new allocation x is depicted in Figure 3.

Loop V: ($F = \{1, 2, 3, 4, 7, 8\} \neq P^+$)

Step 1: $S^1 = \{6\}$.

Step 2: $C^1 = \{6\}$, since player 6 is not adjustable (Definition 3.7 (3)(a) with $k + 1 = 9$). As a consequence, $F = \{1, 2, 3, 4, 5, 6, 7, 8\}$. Since $S^1 \subseteq F$ we go to Loop VI.

Loop VI: $F = \{1, 2, 3, 4, 5, 6, 7, 8\} = P^+$

Hence, we stop and $Lmax(P, v) = x = (0, 3, 7, 3, 1, 2, 2, 4, 0)$. $\square$

In the next lemma we prove that the recursive step is well-defined and that the algorithm does indeed yield the leximax solution. The lemma will also be used to prove that the algorithm terminates in a finite number of steps.

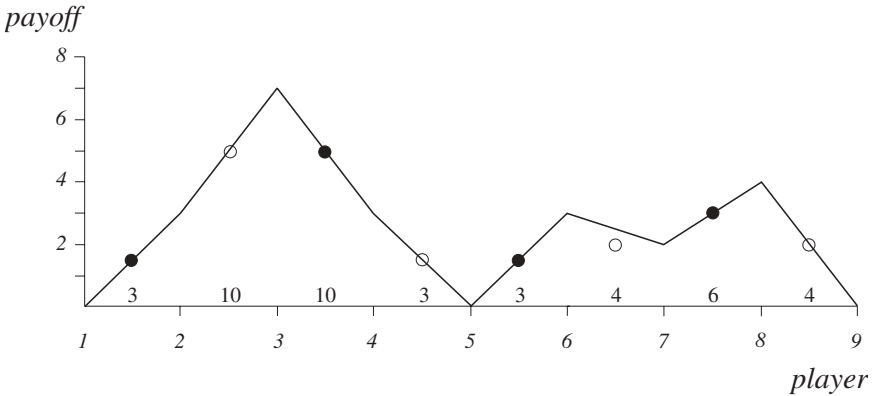


Fig. 2. The allocation x that results from Loop II

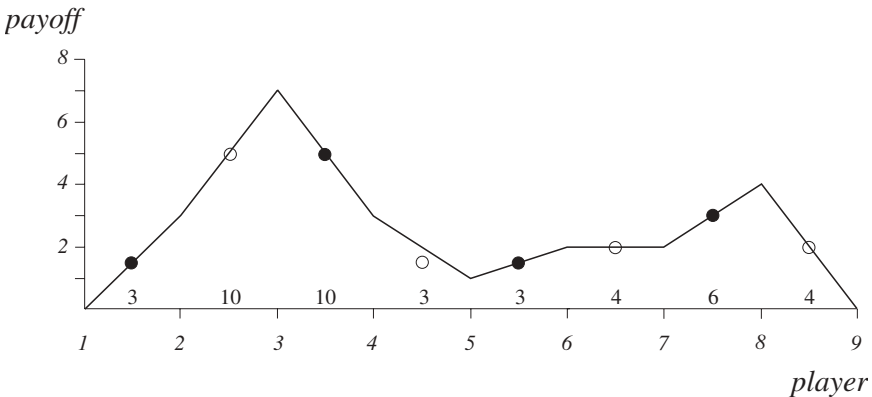


Fig. 3. The allocation x that results from Loop IV

Lemma 4.2. *In the recursive step of the algorithm:*

- (a) *The players that we fix in the F -procedure are inadjustable and remain inadjustable if we do not change the payoffs of the players in F .*
- (b) *We only fix players in P^+ . Moreover, if we fix a player, then we fix his partner too.*
- (c) *If $C^1 \neq \emptyset$, then let $x^* := x_i$ where $i \in C^1$. It holds that $x_i \leq x^*$ for all players $i \notin F$.*
- (d) *In the F -procedure we fix all players in C^1 . Hence, $C^1 \cap S^2 = \emptyset$.*
- (e) *For $\epsilon > 0$ sufficiently small, every player in S^2 satisfies the conditions (B1), (B2), (B3), and (B4).*
- (f) *If $i \in S^2$ and $j \in I(i, x)$, then $j \notin F$.*
- (g) *If $i_1, i_2 \in S^2$ and $i_1 \neq i_2$, then not both $i_1 \in I(i_2, x)$ and $i_2 \in I(i_1, x)$.*
- (h) *The allocation y is well-defined and the payoffs of the fixed players do not change. Moreover, y is a core allocation and $\max_{j \notin F} y_j < \max_{j \notin F} x_j$.*

Proof. The proof is by induction on the number of loops. We assume that (a)–(h) hold for loops $1, \dots, t-1$ of the algorithm and that $F \neq P^+$. Then, we prove that (a)–(h) hold for the t -th loop. The proof of (a)–(h) for the first loop of the algorithm has been omitted, since it is similar to the proof for the t -th loop.

(a) By the induction hypothesis we only have to show that every unfixed player that we fix in the F -procedure is inadjustable by giving a condition in Definition 3.7 that is not satisfied. We distinguish among the three cases in Step 2. Let $i \in C^1$. We may assume, without loss of generality, that $I(i, x) = [i-1, k]$.

Step a. Clearly, $m \geq i$. Let $j \in [i-1, m]$, $j \notin F$.

Suppose $|i-j|$ is even. Then, $j \geq i$ and $I(j, x) = [j-1, k]$. Hence, j is not adjustable by Definition 3.7 (1) and $m \in I(j, x)$.

Suppose $|i-j|$ is odd. Note that $x_j \leq x_i$ (otherwise $i \notin S^1$) and $I(j, x) = [l, j+1]$ for some $l \leq i-1$. Hence, j is not adjustable by Definition 3.7 (2), and $i \in I(j, x)$.

Step b. Clearly, $m^* \geq i-1$. Let $j \in [i-1, m^*+1]$, $j \notin F$.

We have $x_j \leq x_i$ (otherwise $i \notin S^1$).

Suppose $|i-j|$ is even. Note that $x_j \leq x_i \leq x_{m^*}$ and $I(j, x) = [j-1, k]$. Hence, j is not adjustable by Definition 3.7 (2) and $m^* \in I(j, x)$.

Suppose $|i-j|$ is odd. Note that $I(j, x) = [l, j+1]$ for some $l \leq i-1$. Hence, j is not adjustable by Definition 3.7 (2) and $i \in I(j, x)$.

Step c. Let $j \in [i-1, k]$, $j \notin F$.

Suppose $|i-j|$ is even. Then, j is not adjustable by Definition 3.7 (3).

Suppose $|i-j|$ is odd. Note that $x_j \leq x_i$ (otherwise $i \notin S^1$) and $I(j, x) = [l, j+1]$ for some $l \leq i-1$. Hence, j is not adjustable by Definition 3.7 (2) and $i \in I(j, x)$.

As one can verify easily, the discussed unsatisfied conditions above remain unsatisfied in the remainder of the algorithm if we do not change the payoffs of the players in F . Hence, the players that we fix in the F -procedure remain inadjustable in the remainder of the algorithm if we do not change the payoffs of the players in F .

- (b) Follows immediately from the F -procedure.
- (c) Suppose $C^1 \neq \emptyset$. By definition of S^1 , it holds that the payoff of every player in C^1 is the same. So, we can define $x^* := x_i$ for $i \in C^1$. By definition of S^1 , we have that $x_i \leq x^*$ for all players $i \notin F$.
- (d) Let $i \in C^1$. At least one of the conditions for i in Steps a, b, and c in the F -procedure is satisfied. In any case, we fix player i . So, $i \notin S^2$. Hence, $C^1 \cap S^2 = \emptyset$.
- (e) From the definition of C^1 and (d), it follows that each player in S^2 is adjustable. This implies that for $\epsilon > 0$ sufficiently small, every player in S^2 satisfies the conditions (B1), (B2), (B3), and (B4).
- (f) The statement is clear for $j = i$. So, suppose $j \neq i$.

Suppose $j \in F$. By (c), (h), and the induction hypothesis, there exists some player $i_0 \in F$ with $x_{i_0} \geq x_i$, $j \in I(i_0, x)$, and for which all players between i_0 and j are fixed. By (b) and the induction hypothesis, the partner of j in μ is also fixed. One verifies that together with $j \in I(i, x)$ this implies that $i_0 \in I(i, x)$.

If $|i - i_0|$ is odd, then i is not adjustable by Definition 3.7 (2). If $|i - i_0|$ is even, then i is not adjustable for the same reason that i_0 is not adjustable. So, in either case i is not adjustable, contradicting (b). Hence, our assumption that $j \in F$ is false.

- (g) Let $i_1, i_2 \in S^2$ and $i_1 \neq i_2$. Suppose that both $i_1 \in I(i_2, x)$ and $i_2 \in I(i_1, x)$. Then, $|i_1 - i_2|$ is odd. Since $i_1, i_2 \in S^2 \subseteq S^1$, we have $x_{i_1} = x_{i_2}$. So, i_1 and i_2 are not adjustable by Definition 3.7 (2). This contradicts $i_1, i_2 \in S^2$.
- (h) It follows from (g) that y is well-defined. It follows from (f) that the payoffs of fixed players do not change. The inequality $\max_{j \notin F} y_j < \max_{j \notin F} x_j$ follows from the definition of S^1 and the definition of the allocation y . One easily verifies that $y \in \text{Core}(P, v)$ by checking the conditions in Lemma 2.3. We have omitted this part of the proof since it runs similarly to the proof of $y \in \text{Core}(P, v)$ in the ‘only if’-part of the proof of Theorem 3.9. $\square$

The following lemma shows that the algorithm terminates after a finite number of steps.

Lemma 4.3. *After at most $2p + 1$ loops the number of fixed players increases strictly.*

Proof. Consider a loop of the algorithm for which $F \neq P^+$ in Step 1. Let x be the allocation in Step 1. Suppose that the number of fixed players does not increase strictly. In other words, suppose that we do not fix any player in the F -procedure. Then we go to Step 3 with $S^2 = S^1 \neq \emptyset$. We make the following three observations.

Observation 1. If there is an equality in (B1) or (B2) for a player $i \in S^2$, then i will be an inadjustable player in the next loop. Since i remains a player with highest payoff among the non-fixed players, he will be fixed in the next loop.

Observation 2. If there is an equality in (B3) for a player $i \in S^2$, then: either

- 1) i is fixed in the next loop (for sure if $k + 1 \in F$ or $k + 1 \in P^-$) or
- 2) the maximal relevant interval of i becomes strictly larger in the next loop (only possible if $k + 1 \in P^+ \setminus F$),

where $I(i, x)$ is, without loss of generality, assumed to be of the form $[i - 1, k]$. By Lemma 4.2 (g), there can be at most p subsequent loops in which

- 1) we do not fix any player in S^2 and
- 2) in which the maximal relevant interval of a player in S^2 becomes strictly larger.

Observation 3. There can be at most p subsequent loops in which we do not fix any player and in which there is no equality in (B1), (B2), or (B3). This follows since a player j appears at most once in an equality in (B4).

From the three observations one can conclude that after at most $1 + p + p$ subsequent loops we fix a player. This proves the lemma. $\square$

Lemma 4.4. *The algorithm for finding the leximax solution of a neighbor game takes $\mathcal{O}(p^3)$ time.*

Proof. It takes at most $\mathcal{O}(p^2)$ time to calculate a core allocation, which serves as initial allocation for the algorithm (see footnote on the initial allocation in the algorithm). It follows from Lemma 4.3 that the algorithm terminates after at most $p(2p + 1)$ loops. Since each loop takes at most $\mathcal{O}(p)$ time we conclude that the algorithm is of order p^3 . $\square$

References

- Arin J, Iñarra E (1997) Consistency and Egalitarianism: The Egalitarian Set. SEEDS Discussion Paper 163, University of the Basque Country, Bilbao, Spain
- Arin J, Kuipers J, Vermeulen D (1998) An Axiomatic Approach to Egalitarianism in TU-games. IEP Discussion Paper 98-05, Instituto de Economía Pública, University of the Basque Country, Bilbao, Spain
- Curiel I, Potters J, Rajendra Prasad V, Tijs S, Veltman B (1994) Sequencing and Cooperation. *Operations Research* 42:566–568
- Hamers H, Klijn F, Solymosi T, Tijs S, Vermeulen D (2003) On the Nucleolus of Neighbor Games. *European Journal of Operational Research* 146:1–18
- Hamers H, Klijn F, Solymosi T, Tijs S, Villar JP (2002) Assignment Games satisfy the CoMa-property. *Games and Economic Behavior* 38:231–239
- Moulin H (1988) *Axioms of Cooperative Decision Making*. Cambridge University Press, Cambridge
- Owen G (1986) Values of Graph-Restricted Games. *SIAM Journal of Algebraic and Discrete Methods* 7:210–220
- Potters J, Reijnierse H (1995) Γ -Component Additive Games. *International Journal of Game Theory* 24:49–56
- Schmeidler D (1969) The Nucleolus of a Characteristic Function Game. *SIAM Journal on Applied Mathematics* 17:1163–1170
- Shapley L, Shubik M (1972) The Assignment Game I: The Core. *International Journal of Game Theory* 1:111–130
- Sharkey W (1982) Cooperative Games with Large Cores. *International Journal of Game Theory* 11:175–182