

Tilburg University

Neighbour Games and the Leximax Solution

Klijn, F.; Vermeulen, D.; Hamers, H.J.M.; Solymosi, T.; Tijs, S.H.; Pere Villar, J.

Publication date:
1999

[Link to publication in Tilburg University Research Portal](#)

Citation for published version (APA):

Klijn, F., Vermeulen, D., Hamers, H. J. M., Solymosi, T., Tijs, S. H., & Pere Villar, J. (1999). *Neighbour Games and the Leximax Solution*. (CentER Discussion Paper; Vol. 1999-110). Operations research.

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

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Neighbour Games and the Leximax Solution

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

^a*Department of Econometrics and CentER, Tilburg University, The Netherlands*

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

^c*Department of Operations Research, Budapest University of Economic Sciences, Hungary*

^d*Departament d'Economia i d'Història Econòmica, Universitat Autònoma de Barcelona, Spain*

Abstract: Neighbour 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 neighbour 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 neighbour games. After that we provide a polynomially bounded algorithm of order p^3 for calculating the leximax solution (cf. Arin and Iñarra (1997)) of neighbour games, where p is the number of players.

Keywords: Neighbour Games; Leximax Solution; Assignment Games

JEL classification: C71, C78

Running title: Leximax Solution for Neighbour Games

¹Corresponding author. Department of Econometrics and CentER, Tilburg University, P.O. Box 90153, 5000 LE Tilburg, The Netherlands. Tel. (31) 13 466 2733; Fax. (31) 13 466 3280; e-mail: F.Klijn@kub.nl

²This 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).

1 Introduction

In this paper we introduce neighbour games and provide an algorithm to calculate the leximax solution (cf. Arin and Iñarra (1997)) of neighbour games. The following two examples describe situations that give rise to neighbour 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 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 neighbours 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. Neighbour 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 neighbour games. In neighbour games, players are lined up in a one-dimensional queue. In this queue, players can only directly cooperate with one of their neighbours in the queue.

It turns out that the class of neighbour 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, neighbour 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, neighbour games satisfy the CoMa-property, i.e., the core is the convex hull of some marginal vectors (cf. Hamers *et al.* (1999a)).

In this paper we study in detail the leximax solution (cf. Arin and Iñarra (1997)) for neighbour 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 neighbour games is studied in Hamers *et al.* (1999b)).

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 neighbour games is not a subclass of any of the above mentioned classes of games we study the leximax solution for neighbour 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 neighbour 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 neighbour 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 Neighbour games

In this section we introduce the class of neighbour games and present some results on the core of neighbour 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 neighbour 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

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

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

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 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 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 .

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 Reijnders (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 neighbour 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 neighbours in the queue. From this point of view, neighbour games are defined as restricted assignment games: only pairs that are neighbours in the queue can be matched. Formally, let P be the player set of size p . For the sake of convenience we assume that $P = \{1, \dots, p\}$. Let $\sigma : P \rightarrow \{1, \dots, p\}$ be an ordering of the players. Obviously, P can be partitioned in the set M of players i in odd position (i.e., $\sigma(i)$ is odd) and the set N of players in even position (i.e., $\sigma(i)$ is even). Players i and j are called *neighbours* if $|\sigma(i) - \sigma(j)| = 1$. We shall use the (unconventional) notation (i, j) if $\sigma(j) = \sigma(i) + 1$, i.e., (\cdot, \cdot) is used to indicate the order of (neighbouring) players as given by σ . For all pairs (i, j) let $a_{ij} \geq 0$ be given. Then, a *neighbour game* (P, v) is defined by

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

where $\mathcal{N}(Q)$ is the set of *matchings* of the players in Q in which each matching consists only of pairs (i, j) that are neighbours. From now on the word *matching* means a matching of this type. A matching $\mu \in \mathcal{N}(Q)$ is called *optimal* for $Q \subseteq P$ if $\sum_{(i,j) \in \mu} a_{ij} = v(Q)$. It is called *minimal* if $a_{ij} > 0$ for all $(i, j) \in \mu$. Note that $v(i) = 0$ for all $i \in P$ and $v(i, j) = a_{ij}$ for all pairs (i, j) .

For the sake of convenience, we assume henceforth that the players in a neighbour game (P, v) are ordered $1 \prec 2 \prec \dots \prec p$.

Example 2.1 Let $P = \{1, 2, 3, 4\}$ be the player set and let σ describe the order $1 \prec 2 \prec 3 \prec 4$. The pairs that are neighbours with respect to σ are $(1,2)$, $(2,3)$, and $(3,4)$. Hence, all other pairs have a worth equal to zero. Take, for instance, $a_{12} = 10$, $a_{23} = 20$, and $a_{34} = 30$. Then the corresponding neighbour game (P, v) is depicted in Table 2.1. The matching that matches

player 1 with player 2 and player 3 with player 4 is optimal and minimal.

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

Table 2.1: a neighbour game (P, v) .

◇

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

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

Let (P, v) be a neighbour game. Since neighbour games are special assignment games, the results of Shapley and Shubik (1972) on the cores of assignment games apply to the cores of neighbour games. In particular, the cores of neighbour games are not empty. Furthermore, they are determined by the inequalities induced by the one-player and the neighbouring pair coalitions. Henceforth, whenever we speak of a coalition it is a singleton or a neighbouring pair of players.

For an optimal matching μ of P we denote, with a slight abuse of notation, by P^+ the set of players that are matched by μ . Define $P^- := P \setminus P^+$, which will be called the set of *isolated* players. The following Lemma is a straightforward consequence of a result of Shapley and Shubik (1972).

Lemma 2.3 *Let (P, v) be a neighbour game. Let μ be an optimal matching of P . Let $x \in \mathbb{R}^P$. Then, $x \in \text{Core}(v)$ 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, neighbour games do not need to be convex, as follows from the next proposition, which provides a necessary and sufficient condition for the convexity of neighbour games.

Proposition 2.4 *A neighbour 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, note that for any $S \subset T \subset P$ and $k \in P \setminus T$ it holds 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

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

□

Although neighbour games are not convex in general, it follows from Hamers *et al.* (1999a) that they satisfy the CoMa-property. In other words, the core of a neighbour game is the convex hull of some marginal vectors.

3 The leximax solution, a characterization

In this section, we recall the leximax solution, which was introduced by Arin and Iñarra (1997). We characterize the leximax solution 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 $x, y \in \mathbb{R}^p$ for some p , we have that $x \preceq_{lex} y$ if either $x = y$ or there exists a k such that $x_i = y_i$ for $i = 1, \dots, k$ and $x_{k+1} < y_{k+1}$. Further, let $\theta(x)$ be the vector that results when arranging the elements of the vector x in a non-increasing order. Then, for a balanced game (P, v) , Arin and Iñarra (1997) defined the leximax solution $Lmax(v)$ as

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

Arin and Iñarra (1997) showed that the leximax solution is a singleton 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.

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 neighbour 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 neighbour game that neighbour 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 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 neighbour games do not have a large core.

Example 3.2 Let (P, v) be the neighbour 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 $x = (4, 2, 8) \in Core(w)$, but there is no $y \in Core(v)$ such that $y_2 \leq x_2$ (since for all $y \in Core(v)$ we have $y_2 \geq 6 > 2 = x_2$). Hence, the neighbour game (P, v) does not have a large core. \diamond

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

Let (P, v) be a neighbour game. Let μ be a minimal optimal matching. The pairs that are matched by μ are called *essential*, i.e., if $(i, j) \in \mu$, then (i, j) is essential.

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 neighbour game. Let μ be a minimal optimal matching and $x \in Core(v)$ 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 $[i - 1, k]$).

³The s stands for successor.

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

Definition 3.4 Let (P, v) be a neighbour game. Let μ be a minimal optimal matching and $x \in \text{Core}(v)$ 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 Suppose $x \in \text{Core}(v)$.

- (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 $(i, i + 1)$ or $(i - 1, i)$ 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

The maximal relevant interval for a player $i \in P^+$ with respect to a core allocation x is henceforth denoted by $I(i, x)$. Let $|I(i, x)|$ denote the number of players in $I(i, x)$.

Lemma 3.6 For $i \in P^+$, $|I(i, x)| = 2l$ for some $l \geq 1$.

Proof. By Lemma 3.5 (ii) we have that $I(i, x) = [i - 1, k]$ or $I(i, x) = [k, i + 1]$. 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 contradicts 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 The payoff x_i of a player $i \in P^+$ can be adjusted⁵ 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 $x_k + x_{k+1} > v(k, k + 1)$.
- (3)(b). If $I(i, x)$ is of the form $[k, i + 1]$, then either $k - 1$ is non-existent or $x_{k-1} + x_k > v(k - 1, k)$.

⁴The p stands for predecessor.

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

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

Lemma 3.8 *Let $x, y \in \mathbb{R}^p$ with $\theta(x) \neq \theta(y)$ and $\theta(y) \preceq_{lex} \theta(x)$. Let $\sigma : \{1, \dots, p\} \rightarrow P$ be a bijection such that $x_{\sigma(1)} \geq \dots \geq x_{\sigma(p)}$. 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 $r > 1$), and $\theta(y) \preceq_{lex} \theta(x)$, 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

Theorem 3.9 *Let x be a core allocation of a neighbour game (P, v) . Then, $x = Lmax(v)$ if and only if no $i \in P^+$ 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 $y \in Core(v)$ with $\theta(y) \neq \theta(x)$ and $\theta(y) \preceq_{lex} \theta(x)$. 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 $y \in Core(v)$ 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, $j = k$. Then, by Lemma 3.6 we have that $|i - k|$ is even. Hence, by (1) we have that $y_k = x_k - \epsilon$ and $y_{k+1} = x_{k+1}$. 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 (1) 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 $x \in Core(v)$.

(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, $y \in Core(v)$.

Now, we will show that $\theta(y) \preceq_{lex} \theta(x)$ and $\theta(y) \neq \theta(x)$. For this we first prove the following two statements.

(i) if $x_j \geq x_i$, then $x_j \geq y_j$ and

(ii) if $x_j < x_i$, then $y_j < y_i$.

To prove (i), assume that $x_j \geq x_i$. If $j \notin I(i, x)$, then $x_j = y_j$. If $j \in I(i, x)$, then, since i can be adjusted with respect to x , $x_j \geq x_i$ implies $|i - j|$ is even. Hence, $y_j = x_j - \epsilon \leq x_j$.

To prove (ii), assume that $x_j < x_i$. If $j \notin I(i, x)$, then $y_j = x_j < x_i - \epsilon = y_i$ where the inequality follows from (A4) of the choice of ϵ . If $j \in I(i, x)$ and $|i - j|$ is even, then $y_j = x_j - \epsilon < x_i - \epsilon = y_i$. If $j \in I(i, x)$ and $|i - j|$ is odd, then $y_j = x_j + \epsilon < x_i - \epsilon = y_i$ where the inequality follows from (A2) of the choice of ϵ .

Now we will use (i) and (ii) to prove that $\theta(y) \preceq_{lex} \theta(x)$ and $\theta(y) \neq \theta(x)$. Let $J := \{j \in P : y_j \geq y_i \text{ and } y_j \neq x_j\}$. Since $x_i > y_i (= x_i - \epsilon)$ we have that $i \in J$. So, $J \neq \emptyset$. Take $k \in \operatorname{argmax}_{j \in J} y_j$.

Note that if $y_j > y_k$, then $y_j = x_j$. If $y_j = y_k$ and $j \notin J$, then $y_j = x_j$. And finally, if $y_j = y_k$ and $j \in J$, then $y_j < x_j$: suppose not, i.e., suppose that $y_j \geq x_j$. Since $j \in J, y_j \neq x_j$. So, $y_j > x_j$. By (i), $x_j < x_i$. By (ii), $y_j < y_i$. This contradicts $j \in J$.

From the above and $k \in J$ it readily follows that $\theta(y) \preceq_{lex} \theta(x)$ and $\theta(y) \neq \theta(x)$.

Now we will prove the ‘if’-part. Suppose there is a core allocation $y \in Core(v)$ with $\theta(y) \neq \theta(x)$ and $\theta(y) \preceq_{lex} \theta(x)$. Let $\sigma : \{1, \dots, p\} \rightarrow P$ be a bijection such that $x_{\sigma(1)} \geq \dots \geq x_{\sigma(p)}$. We may assume, without loss of generality, that 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 $y \in Core(v)$.

Now we check conditions (1), (2), and (3) of Definition 3.7.

(1). Take $j \in I(\sigma(r), x)$ for which $|j - \sigma(r)|$ is even. From $x_{\sigma(r)} > y_{\sigma(r)}, y \in Core(v)$, 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 $x_{\sigma(r)} > y_{\sigma(r)}$, $y \in \text{Core}(v)$, and condition (2) of Definition 3.3 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 $x, y \in \text{Core}(v)$, Lemma 2.3 (iii), and $m + 1 \in P^-$, 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 (1). 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

The leximax solution of two and three person neighbour games (P, v) can be calculated straightforwardly. It is obvious that if $|P| = 2$ with say $v(1, 2) = a$, then $Lmax(v) = (\frac{a}{2}, \frac{a}{2})$. Now suppose $|P| = 3$. Let $a = v(1, 2)$ and $b = v(2, 3)$. We may assume, without loss of generality, that $a \geq b$. Using Definition 3.7 and Theorem 3.9 one easily verifies that $Lmax(v) = (\frac{a}{2}, \frac{a}{2}, 0)$ if $\frac{a}{2} \geq b$, and $Lmax(v) = (a - b, b, 0)$ if $\frac{a}{2} < b$. The next proposition provides a closed formula for the leximax solution in case there are four players involved.

Proposition 3.10 *Let (P, v) be a 4-person neighbour 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,⁶*

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

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 neighbour games. This algorithm is polynomially bounded of order p^3 in the number of players p . Moreover, a nice feature of the algorithm is that it can be visualized nicely by some pictures showing the process of adjusting and fixing payoffs.

⁶For two numbers $d, e \in \mathbb{R}$ we define $d \vee e := \max\{d, e\}$ and $d \wedge e := \min\{d, e\}$.

Let us start with an algorithm to find $Lmax(v)$ for an arbitrary neighbour game (P, v) . The algorithm 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 all players in P^+ are fixed. The final allocation is the leximax solution, since whenever we fix the payoff of a particular player, that player is no longer adjustable in the remainder of the algorithm.

Algorithm for the leximax solution for neighbour games

Input:

A neighbour game (P, v) .
A core allocation⁷ $x \in Core(v)$.

Initialisation:

Let μ be a minimal optimal matching of the players in P .
Let P^+ be the set of players that are matched by μ .
Set $F := \emptyset$. We call F the set of *fixed* players in P^+ .

Recursive step:

Step 1. If $F = P^+$, then **STOP**, $Lmax(v) = x$. 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 procedure. If $t \leq k$:

Take $i := i_t$.

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

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

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 $F := F \cup [i - 1, m^* + 1]$.

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:

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

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 $F := F \cup [m^* - 1, i + 1]$.

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

⁷A core allocation can for example be obtained by solving a certain linear programming problem (cf. Shapley and Shubik (1972)).

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

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

End of the 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 (1), (2), (3), and (4) 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 $x_j < x_i$.

Calculate the smallest positive number $\epsilon > 0$ for which one of the conditions (1), (2), (3), and (4) becomes an equality for one of the players $i \in S^2$.

Define the vector $y \in \mathbb{R}^P$ 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 $x := y$.

Repeat recursive step.

The next lemma shows that the recursive step is well-defined. The lemma will be used to prove that the algorithm terminates in a finite number of steps.

Lemma 4.1 *In the recursive step of the algorithm:*

- (a) *The players that we fix in Step 2 are inadjustable and remain inadjustable if we do not change the payoffs of the players in F . 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 $i \notin F$.*
- (b) $C^1 \cap S^2 = \emptyset$.
- (c) *If $i \in S^2$ and $j \in I(i, x)$, then $j \notin F$.*
- (d) *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 $i_2 \in I(i_1, x)$.*
- (e) *For $\epsilon > 0$ sufficiently small, every player in S^2 satisfies the conditions (B1), (B2), (B3), and (B4).*
- (f) *The allocation y is well-defined and does not change the payoffs of the fixed players. 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) – (f) hold for loops $1, \dots, t-1$ of the algorithm and that $F \neq P^+$. Then, we prove that (a) – (f) hold for the t -th loop. The proof of (a) – (f) 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 Step 2 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 $I(i, x) = [i-1, k]$.

CASE I. 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 $i \in I(j, x)$.

CASE II.

Clearly, $m^* \geq i-1$. Let $j \in [i-1, m^*+1]$, $j \notin F$. Hence, $x_j \leq x_i$ (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 $i \in I(j, x)$.

CASE III. 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 $i \notin S^1$). Hence, j is not adjustable by Definition 3.7 (2) and $i \in I(j, x)$.

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 $i \notin F$.

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 player that we fix in Step 2 remain inadjustable in the remainder of the algorithm if we do not change the payoffs of the players in F .

(b) Let $i \in C^1$. Then, we are at least in one of the cases I, II, or III. In any case, we fix player i . So, $i \notin S^2$. Hence, $C^1 \cap S^2 = \emptyset$.

(c) The statement is clear for $j = i$. So, suppose $j \neq i$.

Suppose $j \in F$. By (a), (f), and the induction hypothesis, there exists some player $i_0 \in F$ with $x_{i_0} \geq x_i$ and $j \in I(i_0, x)$ for which all players between i_0 and j are fixed, i.e., players in F . Note that $i \leq i_0$, j or $i \geq i_0$, j (otherwise $i \in F$, contradicting $i \in S^2 \subseteq P^+ \setminus F$). This implies together with $j \in I(i, x)$ and $j \in I(i_0, x)$ 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.

(d) Suppose $i_1 \in I(i_2, x)$ 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 $i_1, i_2 \in S^2$.

(e) From the definition of S^2 and (b), 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) It follows from (d) that y is well-defined. It follows from (c) 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 vector y . One easily verifies that $y \in \text{Core}(v)$ by checking the conditions in Lemma 2.3. We have omitted this part of the proof since it runs similarly to the proof of $y \in \text{Core}(v)$ 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 4.2 *After at most $2p$ loops the number of fixed players increases strictly.*

Proof. Let $|F|$ denote the number of players in F . Consider a loop in which $|F|$ does not increase. Let x be the allocation in Step 1 of that loop.

Since $|F|$ does not increase, we do not fix any player in Step 2. Hence, we go to Step 3 with $S^2 = S^1$.

If there is an equality in (1) or (2) for a player $i \in S^2$, then i will be fixed in the next loop in CASE I or CASE II. So, suppose that there are no equalities in (1) and (2) for any player $i \in S^2$.

If there is an equality in (3), then the maximal relevant interval of a player $i \in S^2$ becomes strictly larger. So, by lemma 4.1 (d), there can be at most p loops with equalities in (3) and without equalities in (1) and (2).

Note also that there can be at most p loops with equalities in (4) without equalities in (1) and (2). This follows since a player j appears at most once in an equality in (4).

We can conclude that there are at most $2p$ subsequent loops that only have equalities in (3) and (4). So, after at most $2p$ loops we have an equality in (1) or (2), and thus the number of fixed players $|F|$ increases strictly. \square

Lemma 4.3 *The algorithm for finding the leximax solution of a neighbour games takes $\mathcal{O}(p^3)$ time.*

Proof. It follows from Lemma 4.2 that the algorithm terminates after at most $2p^2$ loops. Since each loop takes at most $\mathcal{O}(p)$ time we have that the algorithm of order p^3 . This proves the Lemma. \square

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

Example 4.4 Consider the neighbour game (P, v) where $P = \{1, \dots, 9\}$ is the set of players, which are ordered $1 \prec \dots \prec 9$. Let v be the characteristic function determined by the values of the neighbours in Table 3.1.

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

Table 3.1: the values of the neighbours in the neighbour game (P, v) .

One readily verifies that there is a unique minimal optimal matching, viz. the matching that matches 1 with 2, 3 with 4, 5 with 6, and 7 with 8. A core allocation is for example $x = (0, 3, 7, 3, 0, 3, 1, 5, 0)$.

We can depict the game and the core allocation x in Figure 3.15. We put the players along the x-axis and their respective payoffs along the y-axis. We connect the payoffs of the players so that the allocation x corresponds to a piece wise linear graph. Moreover, using Lemma 2.3 we immediately see that x is indeed a core allocation:

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

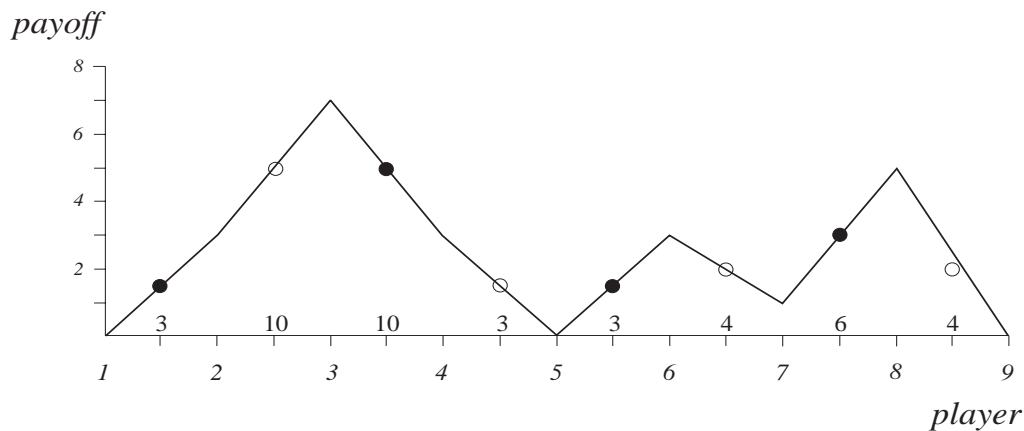


Figure 3.15: The initial allocation x .

We apply the algorithm to find the leximax solution for the game (P, v) . Note $P^+ = \{1, \dots, 8\}$ 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 (1) with $j = 1$). As a consequence, $F = \{1, 2, 3, 4\}$. Since $S^1 \subseteq F$ we go to:

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 3.16.

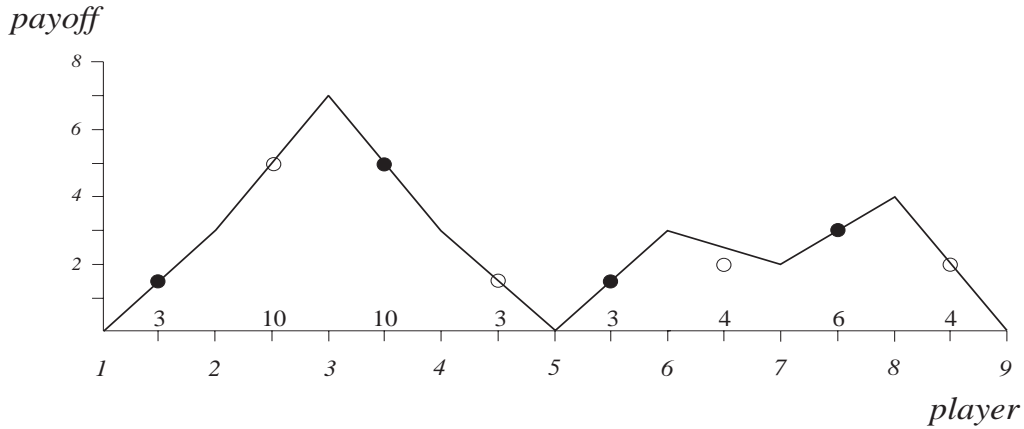


Figure 3.16: The allocation x that results from Loop II.

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

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 3.17.

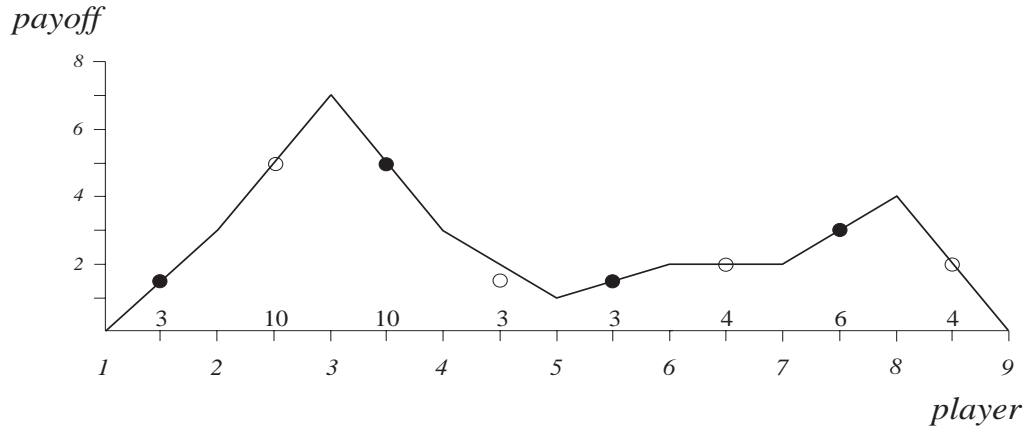


Figure 3.17: The allocation x that results from Loop IV.

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

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

References

- [1] ARIN J. AND IÑARRA E. (1997): "Consistency and Egalitarianism: the Egalitarian Set," Discussion Paper 163, University of País Vasco, Bilbao, Spain.
- [2] ARIN J., KUIPERS J., AND VERMEULEN D. (1998): "An Axiomatic Approach to Egalitarianism in TU-games," Working Paper, University of Maastricht, The Netherlands.
- [3] CURIEL I., POTTERS J., RAJENDRA PRASAD V., TIJS S., AND VELTMAN B. (1994): "Sequencing and Cooperation," *Operations Research*, 42, 566-568.
- [4] HAMERS H., KLIJN F., SOLYMOSI T., TIJS S., AND VILLAR J. (1999a): "On the Extreme Points of the Core of Assignment Games," CentER DP 9943, Tilburg University, The Netherlands.
- [5] HAMERS H., KLIJN F., SOLYMOSI T., TIJS S., AND VERMEULEN D. (1999b): "On the Nucleolus of Neighbour Games," CentER DP 99111, Tilburg University, The Netherlands.
- [6] MOULIN H. (1988): *Axioms of Cooperative Decision Making*, Cambridge University Press, Cambridge, USA.
- [7] OWEN G. (1986): "Values of Graph-Restricted Games," *SIAM Journal of Algebraic and Discrete Methods*, 7, 210-220.
- [8] POTTERS J. AND REIJNIERSE H. (1995): "T-Component Additive Games," *International Journal of Game Theory*, 24, 49-56.
- [9] SCHMEIDLER D. (1969): "The Nucleolus of a Characteristic Function Game," *SIAM Journal on Applied Mathematics*, 17, 1163-1170.
- [10] SHAPLEY L. AND SHUBIK M. (1972): "The Assignment Game I: The Core," *International Journal of Game Theory*, 1, 111-130.