

Article

The Integer Nucleolus of Directed Simple Games: A Characterization and an Algorithm

Reiner Wolff

Department of Economics, University of Fribourg, 1700 Fribourg, Switzerland; reiner.wolff@unifr.ch

Academic Editors: Luca Dall'Asta and Paolo Pin

Received: 16 November 2016; Accepted: 7 February 2017; Published: 28 February 2017

Abstract: We study the class of directed simple games, assuming that only integer solutions are admitted; i.e., the players share a resource that comes in discrete units. We show that the integer nucleolus—if nonempty—of such a game is composed of the images of a particular payoff vector under all symmetries of the game. This payoff vector belongs to the set of integer imputations that weakly preserve the desirability relation between the players. We propose an algorithm for finding the integer nucleolus of any directed simple game with a nonempty integer imputation set. The algorithm supports the parallel execution of multiple threads in a computer application. We also consider the integer prenucleolus and the class of directed generalized simple games.

Keywords: integer nucleolus; integer prenucleolus; desirability relation; simple games

JEL Classification: C63; C71; D72

1. Introduction

The nucleolus is a popular solution concept for cooperative transferable-utility (TU) games because it is a set of game outcomes that are in a sense most acceptable “... as a compromise between the players” (Schmeidler [1], p. 1163). A cooperative TU game is a pair $(N, v) =: \Gamma$, henceforth called a game, with a finite set $N := \{1, \dots, n\}$ of n (≥ 2) players and a coalition function $v : 2^N \rightarrow \mathbb{R}$, $v(\emptyset) = 0$. For every payoff vector $\mathbf{x} \in \mathbb{R}^n$ and coalition $S \subseteq N$, let $x(S) := \sum_{i \in S} x_i$. If $S \subset N$, $S \neq \emptyset$, we denote the excess of coalition S at \mathbf{x} by $e(S, \mathbf{x}) := v(S) - x(S)$. This way, we can associate with each $\mathbf{x} \in \mathbb{R}^n$ the vector $\theta(\mathbf{x})$ of all $2^n - 2$ excesses $e(S, \mathbf{x})$, arranged in nonincreasing magnitude. The nucleolus of Γ with respect to a set X of feasible payoff vectors is the set $\mathcal{N}(\Gamma, X) := \{\mathbf{x} \in X : \theta(\mathbf{x}) \leq_{lex} \theta(\mathbf{z}) \text{ for all } \mathbf{z} \in X\}$ of payoff vectors which minimize $\theta(\mathbf{x})$ lexicographically over X (cf. Schmeidler [1]), as if they were the outcome of an egalitarian arbitration among player coalitions (cf. Maschler [2], p. 611). Our choice for X will be the integer imputation set $I(\Gamma) := \{\mathbf{x} \in \mathbb{Z}^n : x(N) = v(N), x_i \geq v(\{i\}) \text{ for all } i \in N\}$, under the condition that $v(N) \in \mathbb{Z}_{>0}$. Thus, we assume that the players face an integer resource allocation problem in that $v(N)$ comes in discrete units that cannot be broken apart. We call $\nu(\Gamma) := \mathcal{N}(\Gamma, I(\Gamma))$ the integer nucleolus of the game Γ .

A problem closely related to the integer nucleolus is that of finding minimum (or minimum-sum) integer representations of weighted majority games (cf. Peleg [3], Krohn and Sudhölter [4], Kurz et al. [5]). In Wolff and Karagök [6], the integer nucleolus is used to analyze the apportionment of ministries in the Swiss parliament. Fragnelli and Gastaldi [7] study the bankruptcy problem where the estate and the claims are integer, and compare integer versions of the Talmud solution to this problem with the integer nucleolus of the corresponding pessimistic bankruptcy game when there are two or three agents. It is interesting to note that Fragnelli et al. [8,9] suggest integer solutions to a bankruptcy setting that involves an integer estate, while the claims can be of any positive

magnitude. An example is the assignment of emergency intervention units, when the claims are calculated in proportion to the size of the population or area to attend to, which, in turn, means that the characteristic function of an associated bankruptcy game may assume noninteger values. A bankruptcy-based model for the management of capacity-constrained integer resources in mobile radio networks is proposed in Lucas-Estañ et al. [10]. Further examples of integer resources are college places, housing opportunities, livestock, or organs for transplant. Finally, when a sum of money is to be divided among a set of players, each smallest relevant money unit also is a whole object.

Since $I(\Gamma)$ lacks convexity, $\nu(\Gamma)$ may contain multiple elements. The question arises in such a case whether the payoff vectors in $\nu(\Gamma)$ possess some common attributes, beyond the property that they all minimize θ lexicographically. A complementary practical question is how to compute $\nu(\Gamma)$. In general, $\nu(\Gamma)$ can be obtained by solving a sequence of integer linear programs, analogous to a suggestion by Maschler [2] (p. 615) that is implemented in most of the studies compiled by Leng and Parlar [11] (p. 669):

$$\begin{aligned}
 \text{ILP}_k : \quad & \min_{\epsilon, \mathbf{x}} \quad \epsilon \\
 \text{s.t.} \quad & x(S) + \epsilon \geq v(S) \text{ for all } S \in 2^N \setminus \{\mathcal{A}_0, \dots, \mathcal{A}_{k-1}\}, \\
 & x(S) + \epsilon_i = v(S) \text{ for all } S \in \mathcal{A}_i, i = 1, \dots, k-1, \\
 & \mathbf{x} \in I(\Gamma), \\
 & k = 1, 2, \dots,
 \end{aligned} \tag{1}$$

where $\mathcal{A}_0 := \{\emptyset, N\}$, while $\mathcal{A}_i, i \geq 1$, is the set of coalitions with excess $\epsilon_i < \dots < \epsilon_1$ for each optimal solution (ϵ_i, \mathbf{x}) to ILP_i (cf. Guajardo and Jörnsten [12] and Nguyen and Thomas [13]). For some $k = \kappa$, eventually $2^N \setminus \{\mathcal{A}_0, \dots, \mathcal{A}_\kappa\} = \emptyset$, in which case ILP_κ is solved by the set of pairs $(\epsilon_\kappa, \mathbf{x})$ with $\mathbf{x} \in \nu(\Gamma)$. However, this is only a feasible procedure if n is small, since integer linear programs are usually much harder to solve than their continuous counterparts. The present article provides a characterization of $\nu(\Gamma)$ with respect to the class of directed simple games, and suggests an algorithm to compute $\nu(\Gamma)$ for this game class more efficiently, as long as $I(\Gamma)$ is not too large. We focus on directed simple games because they have numerous applications—e.g., in economics, in political science, and in threshold logic (cf. Taylor and Zwicker [14] for literature references).

A simple game Γ_t with $t \in \mathbb{Z}_{>0}$ is a monotonic game Γ , whereby $v(S) \leq v(T)$ if $S \subset T \subseteq N$, that satisfies $v(S) \in \{0, t\}$ for each $S \subseteq N$, while $v(N) = t$. We denote by $\mathcal{W}(\Gamma_t) := \{S \subseteq N : v(S) = t\}$ the set of winning coalitions of Γ_t . A simple game Γ_t is called an ordered (or complete or linear) simple game, if it exhibits a complete desirability relation \succsim_D on the players in the set N . It is called a directed simple game, if in addition $1 \succsim_D \dots \succsim_D n$, without loss of generality. Given a game Γ , if $i \succsim_D j$ for two players $i, j \in N$, then player i is said to be more desirable than player j , in which case

$$v(S \cup \{i\}) \geq v(S \cup \{j\}) \text{ for all } S \subseteq N \setminus \{i, j\}. \tag{2}$$

Player i is said to be strictly more desirable than player j , denoted by $i \succ_D j$, if $i \succsim_D j$, while not $j \succsim_D i$. The players are called symmetric—denoted by $i \sim_D j$ —if both $i \succsim_D j$ and $j \succsim_D i$ (cf. Maschler and Peleg [15], Sect. 9). The desirability relation may be hidden in the set of winning coalitions, and can then be computed with polynomial-time methods (cf. Aziz [16] and Aziz [17], Ch. 3). It may also be rooted naturally in the setup of the game. For example, consider a simple game Γ_t that is a weighted majority game with a quota $q \in \mathbb{R}_{>0}$ and the weights $w_1, \dots, w_n \in \mathbb{R}_{\geq 0}$ of the players, where $q \leq \sum_{i \in N} w_i$. Here, $\mathcal{W}(\Gamma_t) := \{S \subseteq N : \sum_{i \in S} w_i \geq q\}$. Hence, $i \succsim_D j$ for two players $i, j \in N$ whenever $w_i \geq w_j$. We shall represent such a game by the tuple $[q; w_1, \dots, w_n; t]$, in brief $[q; \mathbf{w}; t]$, where $\mathbf{w} := (w_1, \dots, w_n)$. Note that the desirability relation of an ordered simple game Γ_t is strictly preserved by the counting vector $\mathbf{c} := (c_1, \dots, c_n)$, where $c_i := |\{S \subseteq \mathcal{W}(\Gamma_t) : i \in S\}|$ (cf. Lapidot [18,19]). Note also that the integer nucleolus $\nu(\Gamma)$ of any game Γ is nonempty and finite

whenever $I(\Gamma) \neq \emptyset$. A characterization of $v(\Gamma_t)$ will now be derived in Section 2. The algorithm and a performance analysis follow in Sections 3 and 4. Section 5 provides an extension to the integer prenucleolus, and Section 6 concludes.

2. A Characterization of $v(\Gamma_t)$

We begin with two general properties of the integer nucleolus:

Lemma 1. *Let Γ be a game. If $i \succ_D j$ for two players $i, j \in N$, then $x_i \geq x_j$ for each $\mathbf{x} \in v(\Gamma)$.*

Proof. Take any $\mathbf{x} \in v(\Gamma)$. Suppose $x_i \leq x_j - 1$, noting that $x_i \geq v(\{i\})$ by definition of $v(\Gamma)$. Assume $i \succ_D j$, so $v(S \cup \{i\}) \geq v(S \cup \{j\})$ for every $S \subseteq N \setminus \{i, j\}$ by definition. In particular, $v(\{i\}) \geq v(\{j\})$, and thus $x_j - 1 \geq x_i \geq v(\{i\}) \geq v(\{j\})$.

Observe that $e(S \cup \{i\}, \mathbf{x}) \geq e(S \cup \{j\}, \mathbf{x}) + 1$ for each $S \subseteq N \setminus \{i, j\}$. Since $i \succ_D j$ by assumption, there exists a coalition $T \subseteq N \setminus \{i, j\}$ such that $v(T \cup \{i\}) > v(T \cup \{j\})$, implying $e(T \cup \{i\}, \mathbf{x}) \geq e(T \cup \{j\}, \mathbf{x}) + 2$.

Hence, a reallocation at \mathbf{x} of one unit of payoff from player j to player i will result in an integer imputation that improves θ , a contradiction (note that the excesses of all coalitions which contain neither or both of the players i and j are not affected). Therefore, if $i \succ_D j$ and $\mathbf{x} \in v(\Gamma)$, then $x_i \geq x_j$. \square

Lemma 2. *Let Γ be a game. If $i \sim_D j$ for two players $i, j \in N$, then $|x_i - x_j| \in \{0, 1\}$ for each $\mathbf{x} \in v(\Gamma)$.*

Proof. Take any $\mathbf{x} \in v(\Gamma)$. Without loss of generality, suppose $x_i \leq x_j - 2$, noting again that $x_i \geq v(\{i\})$ by definition of $v(\Gamma)$. Assume $i \sim_D j$, so $v(S \cup \{i\}) = v(S \cup \{j\})$ for every $S \subseteq N \setminus \{i, j\}$ by definition. In particular, $v(\{i\}) = v(\{j\})$, and thus $x_j - 2 \geq x_i \geq v(\{i\}) = v(\{j\})$. At the same time, $e(S \cup \{i\}, \mathbf{x}) \geq e(S \cup \{j\}, \mathbf{x}) + 2$ for each $S \subseteq N \setminus \{i, j\}$. Consequently, a reallocation at \mathbf{x} of one unit of payoff from player j to player i yields an integer imputation that improves θ , a contradiction. Therefore, if $i \sim_D j$ and $\mathbf{x} \in v(\Gamma)$, then $|x_i - x_j| \in \{0, 1\}$. \square

Remark 1. *Let Γ be a game, where $i \sim_D j$ for two players $i, j \in N$. If $x_i \neq x_j$ for $\mathbf{x} \in v(\Gamma)$, and payoff vector \mathbf{z} differs from \mathbf{x} only in that the payoffs of players i and j are permuted, then $\mathbf{z} \in v(\Gamma)$ by the anonymity (impartiality) of $v(\Gamma)$.*

The following lemma and a subsequent corollary are our main theoretical result:

Lemma 3. *The integer nucleolus $v(\Gamma_t)$ of a directed simple game Γ_t with $I(\Gamma_t) \neq \emptyset$ has a unique element, henceforth denoted by \mathbf{y} , that weakly preserves \succsim_D ; i.e., $y_1 \geq \dots \geq y_n$.*

Proof. By definition of a directed game, $1 \succsim_D \dots \succsim_D n$. Then, because of Lemmas 1 and 2, Remark 1, and the nonemptiness of $v(\Gamma_t)$, a payoff vector like \mathbf{y} exists. To prove its uniqueness, suppose for $\mathbf{x} \in I(\Gamma_t)$ that $\mathbf{x} \neq \mathbf{y}$ and $x_1 \geq \dots \geq x_n$. Denote by i the largest index j for which $x_j \neq y_j$. Hence, $x_j = y_j$ for $j > i$. Without loss of generality, suppose $x_i > y_i$. Note that $\mathbf{x} \in v(\Gamma_t)$ if and only if $\theta(\mathbf{x}) = \theta(\mathbf{y})$:

Case 1: $\{i\} \notin \mathcal{W}(\Gamma_t)$; i.e., $e(\{i\}, \mathbf{y}) = -y_i$. Assume $y_i = 0$. Then, $x_j = 0$ if and only if $j > i$, in which case also $y_j = 0$. Now consider any coalition $S \subset N$, $S \neq \emptyset$, with $e(S, \mathbf{x}) = 0$. If $S \in \mathcal{W}(\Gamma_t)$, then $x(S) = t$, and thus $x_l = 0$ for every player $l \in N \setminus S$. Since $y_l = 0$ for each such player, $y(S) = t$ and hence $e(S, \mathbf{y}) = 0$. If $S \notin \mathcal{W}(\Gamma_t)$, then $x_k = 0$, and thus $y_k = 0$, for every $k \in S$, whereby also $e(S, \mathbf{y}) = 0$. However, as $e(\{i\}, \mathbf{x}) = -x_i < 0$, we conclude that $\theta(\mathbf{x}) \neq \theta(\mathbf{y})$.

Next, assume $y_i > 0$. Observe for any coalition $S \subset N$, $S \neq \emptyset$, with $e(S, \mathbf{x}) = -y_i$ that $S \subseteq \{i + 1, \dots, n\}$. Hence, $e(S, \mathbf{y}) = -y_i$. Consequently, as $e(\{i\}, \mathbf{x}) = -x_i < -y_i$, again $\theta(\mathbf{x}) \neq \theta(\mathbf{y})$. In all, $\mathbf{x} \notin v(\Gamma_t)$.

Case 2: $\{i\} \in \mathcal{W}(\Gamma_t)$, whereby $y_i = t$ since $\mathbf{y} \in I(\Gamma_t)$. In particular, $i = 1$, as $y_1 \geq \dots \geq y_n$. Hence, $\mathbf{y} = (t, 0, \dots, 0)$, and therefore $\mathbf{x} \notin I(\Gamma_t)$, a contradiction.

To summarize, it cannot be true that $\mathbf{x} \in v(\Gamma_t)$, $\mathbf{x} \neq \mathbf{y}$, and $x_1 \geq \dots \geq x_n$. \square

Remark 2. The proof of Lemma 3 does not depend on the assumed monotonicity of simple games, and the desirability relation \succsim_D was not needed to show the uniqueness of \mathbf{y} . We conclude from the latter observation—since we are free to renumber the players—that if Γ is a simple game, then $v(\Gamma)$ contains at most one payoff vector for every order of the payoffs x_1, \dots, x_n .¹

A permutation π of the player set N of a game Γ such that $v(\pi(S)) = v(S)$ for each $S \subseteq N$ is called a symmetry of Γ . Because of Lemma 3 and Remark 1:

Corollary 1. The integer nucleolus $v(\Gamma_t)$ of a directed simple game Γ_t with $I(\Gamma_t) \neq \emptyset$ is the set of images of \mathbf{y} under all symmetries of Γ_t .

We close this section with three examples and a remark:

Example 1. Let Γ_t be represented by $[33; 15, 15, 10, 10, 4, 4, 4, 2, 1; 30]$. Then, $\mathbf{y} = (7, 7, 4, 4, 3, 2, 2, 1, 0)$. Additionally, $\mathbf{c} = (189, 189, 163, 163, 147, 147, 147, 129, 129)$; i.e., $c_5 = c_6 = c_7$ and $c_8 = c_9$. Since $y_5 > y_6, y_7$, and $y_8 > y_9$, we obtain a set $v(\Gamma_t)$ of six payoff vectors. We provide them in anti-lexicographic order:

$$v(\Gamma_t) = \{(7, 7, 4, 4, 3, 2, 2, 1, 0), (7, 7, 4, 4, 3, 2, 2, 0, 1), (7, 7, 4, 4, 2, 3, 2, 1, 0), \\ (7, 7, 4, 4, 2, 3, 2, 0, 1), (7, 7, 4, 4, 2, 2, 3, 1, 0), (7, 7, 4, 4, 2, 2, 3, 0, 1)\}.$$

Example 2. Let (the non-proper game) Γ_t be represented by $[6; 4, 4, 2, 2, 2, 1; 8]$. Then, $\mathbf{y} = (2, 2, 1, 1, 1, 1)$, while $\mathbf{c} = (30, 30, 26, 26, 26, 23)$. Hence, $v(\Gamma_t) = \{(2, 2, 1, 1, 1, 1)\}$.

Example 3. Consider the game Γ where $n = 3$ and $v(\{1\}) = v(\{2\}) = 1$, $v(\{3\}) = 0$, $v(\{1, 2\}) = 2$, $v(\{1, 3\}) = v(\{2, 3\}) = 1$, $v(\{1, 2, 3\}) = 7$. Note that $1 \sim 2 \succ 3$, whereas $v(\Gamma) = \{(3, 3, 1), (3, 2, 2), (2, 3, 2)\}$. Lemma 3 and Corollary 1 thus cannot be generalized to every directed game Γ .

Remark 3. In Example 2, player 6 is a null player, yet $y_6 = 1$. In general, let Γ be a game, and write as $L(\Gamma) := \{j \in N : v(S) = v(S \cup \{j\}) \text{ for all } S \subseteq N \setminus \{j\}\}$ the set of null players of Γ . Suppose $\mathbf{x} \in I(\Gamma)$ and $x_i \geq 2$ for $i \in L(\Gamma)$. Since $v(N) > 0$, there exists a coalition $T \in \arg \max_{\emptyset \neq S \subseteq N} e(S, \mathbf{x})$, such that $T \cap L(\Gamma) = \emptyset$, whereby $e(Q, \mathbf{x}) \leq e(T, \mathbf{x}) - 2$ for all $Q \subset N$, $i \in Q$. So, a reallocation at \mathbf{x} of one unit of payoff from player i to a player $k \in T$ improves θ . Hence, $x_i \in \{0, 1\}$ for each $i \in L(\Gamma)$ and every $\mathbf{x} \in v(\Gamma)$.

3. An Algorithm to Compute $v(\Gamma_t)$

Given a directed simple game Γ_t , denote by $\theta_{\mathcal{W}}$ the subvector of θ associated with the coalitions in $\mathcal{W}(\Gamma_t) \setminus N$, and by $P(\Gamma_t) := \{\mathbf{x} \in I(\Gamma_t) : x_1 \geq \dots \geq x_n \wedge |x_i - x_j| \in \{0, 1\} \text{ for all } i \neq j, i \sim_D j\}$ the subset of integer imputations in anti-lexicographic order that satisfy Lemmas 1 and 2. Let $v_{\mathcal{W}}(\Gamma_t) := \{\mathbf{x} \in P(\Gamma_t) : \theta_{\mathcal{W}}(\mathbf{x}) \leq_{lex} \theta_{\mathcal{W}}(\mathbf{z}) \text{ for all } \mathbf{z} \in P(\Gamma_t)\}$. Write as $V(\Gamma_t) := \{i \in N : S \in \mathcal{W}(\Gamma_t) \Rightarrow i \in S\}$ the set of veto players of Γ_t , and let $r := |V(\Gamma_t)|$. If $r > 0$, then $V(\Gamma_t) = \{1, \dots, r\}$, because Γ_t is directed.

Lemma 4. Let Γ_t be a directed simple game with $I(\Gamma_t) \neq \emptyset$. If $\mathcal{W}(\Gamma_t) \setminus N \neq \emptyset$, then $v_{\mathcal{W}}(\Gamma_t) = \{\mathbf{y}\}$.

Proof. Assume $\mathcal{W}(\Gamma_t) \setminus N \neq \emptyset$, and note that $P(\Gamma_t) \neq \emptyset$ due to $I(\Gamma_t) \neq \emptyset$. Hence, $v_{\mathcal{W}}(\Gamma_t) \neq \emptyset$. In particular, $\mathbf{y} \in v_{\mathcal{W}}(\Gamma_t)$. Now suppose $r > 0$, whereby $x(V(\Gamma_t)) = t$ for each $\mathbf{x} \in v_{\mathcal{W}}(\Gamma_t)$. Since $i \sim_D j$

¹ This conclusion was brought to the author’s attention through a valuable comment by J. Derks.

for any two players $i, j \in V(\Gamma_t)$, only one element of the set $P(\Gamma_t)$ assigns the payoff t to coalition $V(\Gamma_t)$, this element being

$$\mathbf{y} = (\underbrace{y + 1, \dots, y + 1}_{s \text{ times}}, \underbrace{y, \dots, y}_{r-s \text{ times}}, \underbrace{0, \dots, 0}_{n-r \text{ times}}), \tag{3}$$

where $y := \lfloor \frac{t}{r} \rfloor$ denotes the largest integer that does not exceed $\frac{t}{r}$, and $s := t - ry$. Thus, $v_{\mathcal{W}}(\Gamma_t) = \{\mathbf{y}\}$.

Next, suppose $r = 0$; i.e., $V(\Gamma_t) = \emptyset$. Take any $\mathbf{x} \in P(\Gamma_t)$, $\mathbf{x} \neq \mathbf{y}$, and denote by i the largest index j such that $x_j \neq y_j$. Accordingly, $x_j = y_j$ for all $j > i$. Without loss of generality, assume $x_i > y_i$. As $i \notin V(\Gamma_t)$, there exists a coalition $S \in \mathcal{W}(\Gamma_t) \setminus N$, $i \notin S$, and hence $N \setminus \{i\} \in \mathcal{W}(\Gamma_t)$ by the monotonicity of Γ_t . Consequently, $e(N \setminus \{i\}, \mathbf{y}) = y_i$. This excess may be obtained at \mathbf{x} only by winning coalitions $N \setminus T$ with $T \subseteq \{i + 1, \dots, n\}$, but then also at \mathbf{y} . Since $e(N \setminus \{i\}, \mathbf{x}) = x_i > y_i$, we obtain $\theta_{\mathcal{W}}(\mathbf{x}) \neq \theta_{\mathcal{W}}(\mathbf{y})$, and so $\mathbf{x} \notin v_{\mathcal{W}}(\Gamma_t)$.² \square

The borderline case of a directed simple game Γ_t with $\mathcal{W}(\Gamma_t) \setminus N = \emptyset$ implies that $\mathcal{W}(\Gamma_t) = \{N\} = V(\Gamma_t)$, and thereby $I(\Gamma_t) \neq \emptyset$, as well as $v(\Gamma_t) = P(\Gamma_t) = \{\mathbf{y}\}$ because of the symmetry of all players. The associated payoff vector \mathbf{y} is determined by (3) for $r = n$.

Thus, if Γ_t is a directed simple game and $I(\Gamma_t) \neq \emptyset$, we may search $P(\Gamma_t)$ for \mathbf{y} without reference to the coalitions not contained in $\mathcal{W}(\Gamma_t)$. In view of this observation, we now suggest an algorithm to compute the integer nucleolus of any such game. Since the elements of $P(\Gamma_t)$ have independent excess vectors $\theta_{\mathcal{W}}$, our algorithm can make the most of computing machinery that supports the parallel execution of multiple threads in an application program. Note for each $\mathbf{x} \in P(\Gamma_t)$ with $x_k > 0$, $x_{k+1} = 0$, that $x_1 + \dots + x_k$ is a partition of t into k ($\leq \min\{t, n\}$) parts in standard form. A fast generator of integer partitions in standard form and in anti-lexicographic order is procedure ZS1 by Zoghbi and Stojmenović [20] (pp. 325–326).

To proceed, define the incidence vector $\mathbf{1}_S \in \{0, 1\}^n$ of any coalition $S \in \mathcal{W}(\Gamma_t) \setminus N$ in that $1_{S_i} = 1$ if and only if $i \in S$, and consider the auxiliary vector $\mathbf{a}_S := (a_{S1}, \dots, a_{Sn})$, where

$$a_{Si} := \sum_{j \leq i} 1_{Sj}, \quad i = 1, \dots, n. \tag{4}$$

Hence, if $\mathbf{x} \in P(\Gamma_t)$ and $x_k > 0$, $x_{k+1} = 0$, then a_{Sk} players in coalition S receive a positive payoff. These players can be read off the auxiliary vector $\mathbf{b}_S := (b_{S1}, \dots, b_{S|S|})$, where

$$b_{Si} := \text{index of the } i\text{th positive entry of } \mathbf{1}_S, \quad i = 1, \dots, |S|. \tag{5}$$

The total payoff of coalition S at \mathbf{x} thus amounts to $\sum_{i \leq a_{Sk}} x_{b_{Si}}$. For example, if $n = 7$ and $S = \{2, 3, 6\}$, we obtain $\mathbf{1}_S = (0, 1, 1, 0, 0, 1, 0)$, $\mathbf{a}_S = (0, 1, 2, 2, 2, 3, 3)$, and $\mathbf{b}_S = (2, 3, 6)$. Thereby, if $t = 6$ and $\mathbf{x} = (2, 2, 1, 1, 0, 0, 0)$ —noting that \mathbf{x} results from a partition of t into four parts— $x(\{2, 3, 6\}) = \sum_{i \leq a_{S4}} x_{b_{Si}} = x_2 + x_3$.

For each $\mathbf{x} \in P(\Gamma_t)$, let $\mathcal{D}(\Gamma_t, \mathbf{x}, h) := \{S \in \mathcal{W}(\Gamma_t) \setminus N : e(S, \mathbf{x}) = h\}$ with $h \in \{0, \dots, t\}$, and attach to \mathbf{x} the auxiliary vector $\Theta(\mathbf{x}) := (\Theta_1(\mathbf{x}), \dots, \Theta_{t+1}(\mathbf{x}))$, where

$$\Theta_{t-h+1}(\mathbf{x}) := |\mathcal{D}(\Gamma_t, \mathbf{x}, h)|, \quad h = t, \dots, 0. \tag{6}$$

This way, $\Theta(\mathbf{x})$ holds the number of occurrences of excesses of a size at \mathbf{x} , top-down from the largest excess t to the smallest excess 0 of the coalitions in the set $\mathcal{W}(\Gamma_t) \setminus N$. Thereby, given any two payoff vectors $\tilde{\mathbf{x}}, \hat{\mathbf{x}} \in P(\Gamma_t)$, $\tilde{\mathbf{x}}$ is preferred to $\hat{\mathbf{x}}$ if and only if $\Theta(\tilde{\mathbf{x}}) <_{lex} \Theta(\hat{\mathbf{x}})$. While calculating and counting

² The author thankfully acknowledges a helpful conversation with J. Derks who pointed out to him that, given a simple game Γ_t , the set $\{\mathbf{x} \in I(\Gamma_t) : \theta_{\mathcal{W}}(\mathbf{x}) \leq_{lex} \theta_{\mathcal{W}}(\mathbf{z}) \text{ for all } \mathbf{z} \in I(\Gamma_t)\}$ contains at most one element \mathbf{x} with $x_1 \geq \dots \geq x_n$, whenever $N \setminus \{i\} \in \mathcal{W}(\Gamma_t)$ for all $i \in N$. We add that it also suffices to impose the weaker condition that $r \leq 1$.

excesses hand in hand, we may be able to tell quite early how $\mathbf{x} \in P(\Gamma_t)$ compares to a current best status quo, say \mathbf{x}^* . Let $\Theta^* := \Theta(\mathbf{x}^*)$, and suppose l is the first index i for which $\Theta_i^* > 0$. Then, payoff vector \mathbf{x} can already be discarded if any of the first $l - 1$ entries of $\Theta(\mathbf{x})$ has been detected to take a positive value or once $\Theta_l(\mathbf{x})$ is known to exceed Θ_l^* .

Table 1. Lexicographic minimization of $\theta(\mathbf{x})$ over $P(\Gamma_t)$.

Input:	Directed simple game Γ_t (assuming $I(\Gamma_t) \neq \emptyset$), p
Output:	(\mathbf{c}, \mathbf{y})
<hr/>	
PROG LEXMIN	
GLOBAL $\mathcal{W}(\Gamma_t) \setminus N, P(\Gamma_t), \mathbf{x}^*(\cdot), p, t$; GLOBAL $\mathbf{a}_S, \mathbf{b}_S$ for all $S \in \mathcal{W}(\Gamma_t) \setminus N$	
$n \leftarrow N $; $t \leftarrow v(N)$	
FOR $i = 1$ TO n DO $\{c_i \leftarrow \{S \subseteq \mathcal{W}(\Gamma_t) : i \in S\} \}$	
$P(\Gamma_t) \leftarrow \{\mathbf{x} \in I(\Gamma_t) : x_1 \geq \dots \geq x_n \wedge x_i - x_j \in \{0, 1\} \text{ for all } i \neq j, c_i = c_j\}$	
FOR each $S \in \mathcal{W}(\Gamma_t) \setminus N$ DO $\{\text{compute } (\mathbf{a}_S, \mathbf{b}_S) \text{ as in (4) and (5)}\}$	
DIM $\mathbf{x}^*(1:p)$	
START THREAD LEXMIN1; ...; START THREAD LEXMIN p	
DO $\{\text{wait for threads LEXMIN1, } \dots, \text{ LEXMIN}p \text{ to complete}\}$	
$\mathbf{y} \leftarrow \mathbf{x}^*(1)$; $\Theta^* \leftarrow \Theta(\mathbf{y})$	
FOR $i = 2$ TO p DO $\{\text{IF } \Theta(\mathbf{x}^*(i)) <_{lex} \Theta^* \text{ THEN } \mathbf{y} \leftarrow \mathbf{x}^*(i)$; $\Theta^* \leftarrow \Theta(\mathbf{y})\}$	
END PROG	
<hr/>	
THREAD FUNCTION LEXMIN i	
$\mathbf{x}^*(i) \leftarrow \text{last } \mathbf{x} \in P(\Gamma_t)$; $\Theta^* \leftarrow \Theta(\mathbf{x}^*(i))$; $l \leftarrow \text{smallest index } j \text{ for which } \Theta_j^* > 0$	
FOR the i th to the 2nd last $\mathbf{x} \in P(\Gamma_t)$ STEP p DO $\{$	
$k \leftarrow \text{number of parts of } t \text{ in } \mathbf{x}$; $\Theta \leftarrow (0, \dots, 0)$	
FOR each $S \in \mathcal{W}(\Gamma_t) \setminus N$ DO $\{$	
$h \leftarrow t - \sum_{j \leq a_{sk}} x_{b_{sj}}$	
IF $t - h + 1 \geq l$ THEN $\Theta_{t-h+1} \leftarrow \Theta_{t-h+1} + 1$ ELSE EXIT FOR	
IF $\Theta_l > \Theta_l^*$ THEN EXIT FOR $\}$	
IF $\Theta <_{lex} \Theta^*$ THEN $\mathbf{x}^*(i) \leftarrow \mathbf{x}$; $\Theta^* \leftarrow \Theta$; $l \leftarrow \text{smallest index } j \text{ for which } \Theta_j^* > 0\}$	
END FUNCTION	

This observation has been built into the computer pseudocode in Table 1. The code distributes most of the workload across a user-defined number p (≥ 1) of threads, in support of machines that can execute multiple threads in parallel (desktop applications will most often benefit from creating as many parallel threads as possible, unless $|P(\Gamma_t)|$ is small). On input, Γ_t and p are passed to the main program LEXMIN that at first generates both n and t as well as the counting vector \mathbf{c} , the set $P(\Gamma_t)$, and the pairs $(\mathbf{a}_S, \mathbf{b}_S)$, $S \in \mathcal{W}(\Gamma_t) \setminus N$. Then, it allocates memory for an array $\mathbf{x}^*(1:p)$ of p best status-quo payoff vectors, before invoking the thread functions LEXMIN1, ..., LEXMIN p . Each of them refers to a separate piece of code, such as the code for thread function LEXMIN i . This function initializes $\mathbf{x}^*(i)$, Θ^* , and l , in terms of a most equal allocation of t across the players, and then enters into two nested loops. In an attempt to spread the computational burden evenly across the threads, the outer loop has been designed to pass over every p th element of the (ordered) set $P(\Gamma_t)$, starting from element number i . For each resulting payoff vector $\mathbf{x} \in P(\Gamma_t)$ and underlying number k of parts of t , the inner loop computes the excesses associated with \mathbf{x} and populates $\Theta(\mathbf{x})$. If in the course of these operations index $t - h + 1$ comes out less than l , or Θ_l^* is exceeded, then \mathbf{x} will be discarded

by a subsequent exit instruction. In the remaining cases, whenever $\Theta(\mathbf{x})$ is lexicographically smaller than Θ^* , the elements of the set $\{\mathbf{x}^*(i), \Theta^*, l\}$ are updated. Once all threads are completed, LEXMIN minimizes Θ lexicographically over the set of retained payoff vectors $\{\mathbf{x}^*(1), \dots, \mathbf{x}^*(p)\}$ in order to obtain the unique argument \mathbf{y} , and then the pair (\mathbf{c}, \mathbf{y}) is returned. Note that all relevant variables are considered as local to LEXMIN and its thread functions unless stated otherwise.

Remark 4. A payoff vector $\mathbf{x} \in P(\Gamma_t)$ can already be discarded if Θ_l is seen to be no smaller than Θ_l^* , while Θ_{l+1} is known to be greater than Θ_{l+1}^* , etc. This may serve to reject \mathbf{x} more quickly, while the extra comparisons can have the opposite effect.

Remark 5. If a reallocation of a unit of payoff between players i and j is unfavorable, another such reallocation can be rejected, everything else being equal. We then do not have to trace Θ over the full set $P(\Gamma_t)$, at the cost of an extended bookkeeping.

Remark 6. One may want to exploit the occurrence of special categories of players (e.g., null players or vetoers), which imposes a search cost (cf. Aziz [16] and Aziz [17], Ch. 3, for the complexity of detecting player types).

Remark 7. The coalitions $S \in \mathcal{W}(\Gamma_t) \setminus N$ have independent pairs $(\mathbf{a}_S, \mathbf{b}_S)$, which suggests a distributed execution of the second loop in LEXMIN.

Once \mathbf{y} is known, we conduct a scan of the counting vector \mathbf{c} for all subvectors (c_i, \dots, c_j) , where $c_i = \dots = c_j$ and $y_i > y_j$, the range $j - i$ being as large as possible. If no such subvector exists, then $v(\Gamma_t) = \{\mathbf{y}\}$, and we are done. Otherwise, given whichever pair (i, j) of matching indices, we retain all elements of $v(\Gamma_t)$ obtained earlier, as well as every payoff vector that can be derived from these elements by permuting the payoffs of the players i, \dots, j .

Table 2. Completion of $v(\Gamma_t)$.

Input:	Directed simple game Γ_t (assuming $I(\Gamma_t) \neq \emptyset$) and associated pair (\mathbf{c}, \mathbf{y})
Output:	$v(\Gamma_t)$
<hr/> PROG COMPL $n \leftarrow N ; \mathbf{y}^1 \leftarrow \mathbf{y}; v(\Gamma_t) \leftarrow \{\mathbf{y}^1\}$ $i \leftarrow n + 1; m \leftarrow 1$ WHILE $i > 2$ DO { $j \leftarrow i - 1; i \leftarrow j$ WHILE $i > 1$ and $c_{i-1} = c_j$ DO $\{i \leftarrow i - 1\}$ IF $y_i > y_j$ THEN { $l \leftarrow 0$ FOR each $\mathbf{b} \in B(i, j)$ DO { FOR $k = 1$ TO m DO { $l \leftarrow l + 1; \mathbf{y}^l \leftarrow (y_1^k, \dots, y_{i-1}^k, \mathbf{b}, y_{j+1}^k, \dots, y_n^k)\}$ $m \leftarrow l; v(\Gamma_t) \leftarrow \{\mathbf{y}^1, \dots, \mathbf{y}^m\}\}$ } } } END PROG	

The pseudocode program COMPL in Table 2 generates a corresponding sequence of approximations $\{\mathbf{y}^1, \dots, \mathbf{y}^m\}$ to $v(\Gamma_t)$. On input, Γ_t and the pair (\mathbf{c}, \mathbf{y}) are passed. To begin with, the program determines n and initializes $v(\Gamma_t)$. Then, starting from $i = j = n$, the program looks for

the smallest $i (< j)$ for which both $c_i = c_j$ and $y_i > y_j$.³ Whenever a matching i is found, the program retrieves all permutations of the payoffs in (y_i, \dots, y_j) from the respective permutation set, named $B(i, j)$, and updates $v(\Gamma_t)$. In the following iteration, j is set to $i - 1$, and so on. The search ends if $i \leq 2$, and then $v(\Gamma_t)$ is returned. Note that for each matching pair (i, j) the respective subvector (y_i, \dots, y_j) of \mathbf{y} takes the form $(a, \dots, a, b, \dots, b)$, where $b = a - 1$. The set $B(i, j)$ can thus be calculated with ACM Algorithm 152 (Nexcom) by Hopley [21], under the normalization $a = 1$.

We end this section with a working example of our algorithm and a generalization. The example is primarily meant to illustrate the functioning of a thread in Table 1. Thus, we assume, for simplicity, that $p = 1$:

Example 4. Let Γ_t be represented by $[9; 5, 4, 3, 2, 1; 6]$. In this game,

$$\mathcal{W}(\Gamma_t) \setminus N = \{ \{1, 2\}, \{1, 2, 3\}, \{1, 2, 4\}, \{1, 2, 5\}, \{1, 3, 4\}, \{1, 3, 5\}, \{2, 3, 4\}, \{1, 2, 3, 4\}, \\ \{1, 2, 3, 5\}, \{1, 2, 4, 5\}, \{1, 3, 4, 5\}, \{2, 3, 4, 5\} \}.$$

Thereby, $\mathbf{c} = (11, 10, 9, 8, 7)$; i.e., the game exhibits the desirability relation $1 \succ_D 2 \succ_D 3 \succ_D 4 \succ_D 5$. Hence,

$$P(\Gamma_t) = \{ (6, 0, 0, 0, 0), (5, 1, 0, 0, 0), (4, 2, 0, 0, 0), (4, 1, 1, 0, 0), (3, 3, 0, 0, 0), (3, 2, 1, 0, 0), \\ (3, 1, 1, 1, 0), (2, 2, 2, 0, 0), (2, 2, 1, 1, 0), (2, 1, 1, 1, 1) \}.$$

The sole thread LEXMIN1 then proceeds as follows if the elements of $\mathcal{W}(\Gamma_t) \setminus N$ arrive at the thread's inner loop in the above order from coalition $\{1, 2\}$ to coalition $\{2, 3, 4, 5\}$:

\mathbf{x}	last Θ in inner loop	$\mathbf{x}^*(1)$	Θ^*	l	last S in inner loop
—	—	(2, 1, 1, 1, 1)	(0, 0, 0, 2, 6, 4, 0)	4	—
(6, 0, 0, 0, 0)	(1, 0, 0, 0, 0, 6)	(2, 1, 1, 1, 1)	(0, 0, 0, 2, 6, 4, 0)	4	{2, 3, 4}
(5, 1, 0, 0, 0)	(0, 1, 0, 0, 0, 2, 4)	(2, 1, 1, 1, 1)	(0, 0, 0, 2, 6, 4, 0)	4	{2, 3, 4}
(4, 2, 0, 0, 0)	(0, 0, 1, 0, 2, 0, 4)	(2, 1, 1, 1, 1)	(0, 0, 0, 2, 6, 4, 0)	4	{2, 3, 4}
(4, 1, 1, 0, 0)	(0, 0, 1, 0, 0, 5, 1)	(2, 1, 1, 1, 1)	(0, 0, 0, 2, 6, 4, 0)	4	{2, 3, 4}
(3, 3, 0, 0, 0)	(0, 0, 0, 3, 0, 0, 4)	(2, 1, 1, 1, 1)	(0, 0, 0, 2, 6, 4, 0)	4	{2, 3, 4}
(3, 2, 1, 0, 0)	(0, 0, 0, 2, 3, 4, 3)	(3, 2, 1, 0, 0)	(0, 0, 0, 2, 3, 4, 3)	4	{2, 3, 4, 5}
(3, 1, 1, 1, 0)	(0, 0, 0, 2, 3, 6, 1)	(3, 2, 1, 0, 0)	(0, 0, 0, 2, 3, 4, 3)	4	{2, 3, 4, 5}
(2, 2, 2, 0, 0)	(0, 0, 0, 0, 9, 0, 3)	(2, 2, 2, 0, 0)	(0, 0, 0, 0, 9, 0, 3)	5	{2, 3, 4, 5}
(2, 2, 1, 1, 0)	(0, 0, 0, 1, 3, 2, 0)	(2, 2, 2, 0, 0)	(0, 0, 0, 0, 9, 0, 3)	5	{1, 3, 5}

Upon initialization, $\mathbf{x}^*(1) = (2, 1, 1, 1, 1)$, whereby $\theta_{\mathbf{y}} = (3, 3, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1)$, such that $\Theta^* = (0, 0, 0, 2, 6, 4, 0)$ and $l = 4$. The first payoff vector $\mathbf{x} = (6, 0, 0, 0, 0)$ in the (ordered) set $P(\Gamma_t)$ will be discarded as soon as LEXMIN1 has calculated the excess of coalition $\{2, 3, 4\}$, since $h = e(\{2, 3, 4\}, \mathbf{x}) = 6$, and thus $t - h + 1 = 1 < 4$, etc. Eventually, Θ^* can be improved at $\mathbf{x} = (3, 2, 1, 0, 0)$, where $\Theta^* = (0, 0, 0, 2, 3, 4, 3)$, and again at $\mathbf{x} = (2, 2, 2, 0, 0)$, where $\Theta^* = (0, 0, 0, 0, 9, 0, 3)$. The latter payoff vector is finally identified as \mathbf{y} by the main program LEXMIN, and the pair (\mathbf{c}, \mathbf{y}) is returned. At last, routine COMPL finds in view of $\mathbf{c} = (11, 10, 9, 8, 7)$ that there is no matching pair (i, j) . Therefore, on output, $v(\Gamma_t) = \{(2, 2, 2, 0, 0)\}$.

For a generalized simple game Γ_t , the monotonicity property is dropped, and it is not maintained that $v(\emptyset) = 0$ and $v(N) = t$ (cf. Carreras and Freixas [22], p. 153, and Taylor and Zwicker [14], p. 4). The desirability relation of a directed generalized simple game is also strictly preserved by the counting vector \mathbf{c} . However, $I(\Gamma_t)$ may have to be redefined as denoting the set $\{ \mathbf{x} \in \mathbb{Z}^n : x(N) = t, x_i \geq v(\{i\}) \text{ for all } i \in N \}$. The proof of Lemma 3 then remains valid, and hence statements analogous to Lemma 3 and Corollary 1 apply. The following result replaces Lemma 4:

³ As to the inner WHILE/DO loop in Table 2, we assume short-circuit evaluation of relational expressions. This means that the expression ' $i > 1$ and $c_{i-1} = c_j$ ' is marked to be false as soon as $i = 1$. Hence, no attempt will be made to access a nonexistent c_0 .

Lemma 5. Let Γ_t be a directed generalized simple game with $I(\Gamma_t) \neq \emptyset$. Suppose $\mathbf{x}, \bar{\mathbf{x}} \in P(\Gamma_t)$, $\mathbf{x} \neq \bar{\mathbf{x}}$. Denote by i the largest index j for which $x_j \neq \bar{x}_j$, where $x_i > \bar{x}_i$ without loss of generality. Then $\theta_{\mathcal{W}}(\mathbf{x}) = \theta_{\mathcal{W}}(\bar{\mathbf{x}}) \Rightarrow \theta(\mathbf{x}) <_{lex} \theta(\bar{\mathbf{x}})$.

Proof. Assume $\{i\} \in \mathcal{W}(\Gamma_t)$, whereby $i = 1$, since Γ_t is directed and $I(\Gamma_t) \neq \emptyset$. Hence, $x_j = \bar{x}_j$ for each player $j \geq 2$. Since $x_1 > \bar{x}_1$, it cannot be true that both \mathbf{x} and $\bar{\mathbf{x}}$ are contained in $P(\Gamma_t)$. So, assume $\{i\} \notin \mathcal{W}(\Gamma_t)$. Consider any nonempty coalition $S \subset N$, $S \notin \mathcal{W}(\Gamma_t)$. Observe that $x_j = \bar{x}_j$ whenever $j > i$. Consequently, if $e(S, \mathbf{x}) > -\bar{x}_i$ or $e(S, \bar{\mathbf{x}}) > -\bar{x}_i$, and thus $S \subseteq \{i + 1, \dots, n\}$, then $e(S, \mathbf{x}) = e(S, \bar{\mathbf{x}})$. Furthermore, if $e(S, \mathbf{x}) = -\bar{x}_i$, and hence again $S \subseteq \{i + 1, \dots, n\}$, then also $e(S, \bar{\mathbf{x}}) = -\bar{x}_i$. Therefore, and since $e(\{i\}, \mathbf{x}) = -x_i < -\bar{x}_i = e(\{i\}, \bar{\mathbf{x}})$, if $\theta_{\mathcal{W}}(\mathbf{x}) = \theta_{\mathcal{W}}(\bar{\mathbf{x}})$, then $\theta(\mathbf{x}) <_{lex} \theta(\bar{\mathbf{x}})$. \square

Hence, if $\theta_{\mathcal{W}}(\mathbf{x}) = \theta_{\mathcal{W}}(\bar{\mathbf{x}})$ for two payoff vectors $\mathbf{x}, \bar{\mathbf{x}} \in P(\Gamma_t)$, we can always discriminate between them by inspecting the smallest payoff components by which they differ. The required adjustment of the code in Table 1 is straightforward.

4. Performance Analysis

The computational workload of the algorithm in Section 3 is mostly determined by the cost of finding \mathbf{y} . In a worst-case scenario, if $n \geq t$ while there are no symmetric players, then $|P(\Gamma_t)| \sim \frac{1}{4t\sqrt{3}} e^{\pi\sqrt{\frac{2t}{3}}} = O(\frac{1}{t} e^{\sqrt{t}})$ (cf. Hall [23], p. 44) for the asymptotic total number of payoff vectors that have to be considered. In another worst-case scenario, every coalition $S \subseteq N$, $S \notin \{\emptyset, \{2\}, \dots, \{n\}\}$, is winning, and the inner loop of each thread function then executes $|\mathcal{W}(\Gamma_t) \setminus N| = 2^n - n - 1 = O(2^n)$ times altogether.

The algorithm has been applied on two personal computers to solve several samples of weighted majority games Γ_t , where $w := \sum_{i=1}^n w_i$ is an odd sum of integer weights, and $q = \frac{w+1}{2}$. The weights have been drawn from the interval $[1, n]$ at random, in terms of uniform deviates as supplied by a system routine. The chosen order of the coalitions $S \in \mathcal{W}(\Gamma_t) \setminus N$ in the thread functions was from the smallest to the largest coalitions, with a lexicographic ordering of each group of coalitions of a size (cf. Example 4). One computer (machine 1) was equipped with a dual-core CPU (Intel(R) Core(TM) i7-2620M, 2.70 GHz) and the other (machine 2) with a quad-core CPU (Intel(R) Core(TM) i5-760, 2.80 GHz). Both CPUs possess four logical processors, whereby $p = 4$. The algorithm’s run-times were measured as the elapsed time when computing the solution of a given game from the pair (n, t) and the set $\mathcal{W}(\Gamma_t) \setminus N$. Aiming at a performance improvement (albeit small), the main program LEXMIN in Table 1 was modified to take account of the suggestion in Remark 7.

Two samples of fifty games have been solved with machine 1. In the first sample, $(n, t) = (7, 300)$, whereas $(n, t) = (10, 100)$ in the second. The average run-times per game were about four seconds (first sample) and about five seconds (second sample). Two samples of ten games have been solved with machine 2. In the first of these samples, $(n, t) = (20, 30)$, while $(n, t) = (20, 40)$ in the second. The average run-times were about two seconds (first sample) and about ten seconds (second sample). In comparison, the Gurobi(TM) [24] linear-programming tool (version 6.5.0, with default settings) needed between about 4 and 35 min, with an average of about $20\frac{1}{2}$ min, to complete only the first linear program of procedure (1) on machine 2 for the ten games where $(n, t) = (20, 30)$. At this stage, just one element of a single game’s integer nucleolus (of cardinality three) could be found. The tool used all available cores of the CPU.

5. Extension to the Integer Prenucleolus

If a simple game Γ_t admits two or more winning single-player coalitions, then $v(\Gamma_t) = I(\Gamma_t) = \emptyset$. In such a case—or when the individual rationality of the game solution is not an issue—we may refer to the integer prenucleolus $\nu^*(\Gamma_t) := \mathcal{N}(\Gamma_t, I^*(\Gamma_t))$, where $I^*(\Gamma_t) := \{\mathbf{x} \in \mathbb{Z}^n : x(N) = v(N)\}$ is the game’s integer preimputation set with $v(N) = t \in \mathbb{Z}_{>0}$. Note that $\nu^*(\Gamma_t)$ is nonempty and finite.

Lemma 6. Let Γ_t be a simple game. If $\mathbf{x} \in v^*(\Gamma_t)$, then $-1 \leq x_i \leq t$ for all $i \in N$.

Proof. Take any $\mathbf{x} \in v^*(\Gamma_t)$ and let $N_- := \{j \in N : x_j \leq -1\}$. Observe that $\mathbf{x} \in I^*(\Gamma_t)$, whereby $N \setminus N_- \neq \emptyset$. Then, consider an arbitrary player $i \in N$:

(a) Assume $x_i \leq -2$. Hence, $i \in N_-$. Let $Q := \arg \max_{\emptyset \neq S \subseteq N} e(S, \mathbf{x})$ and consider any coalition $Q \in \mathcal{Q}$, whereby $|Q| < n$. Since Γ_t is monotonic, $N_- \subseteq Q$, and hence there must be a player $k \in N \setminus N_-, k \notin Q$. Now take any coalition $T \in \arg \max_{k \in S \subseteq N \setminus \{i\}} e(S, \mathbf{x})$. Suppose $T = N \setminus \{i\}$, such that $e(T, \mathbf{x}) = v(N \setminus \{i\}) - x(N \setminus \{i\}) \leq t - (x(N) - x_i) = x_i \leq -2$. Since $e(\{i\}, \mathbf{x}) \geq 2$, a reallocation at \mathbf{x} of one unit of payoff from player k to player i improves θ , and therefore $\mathbf{x} \notin v^*(\Gamma_t)$. Next, suppose $T \subset N \setminus \{i\}$. Observe that $e(T, \mathbf{x}) \leq e(T \cup \{i\}, \mathbf{x}) + x_i$ by the monotonicity of Γ_t , and that $e(T \cup \{i\}, \mathbf{x}) \leq e(Q, \mathbf{x})$ by definition of the set \mathcal{Q} . Consequently, $e(T, \mathbf{x}) \leq e(Q, \mathbf{x}) + x_i \leq e(Q, \mathbf{x}) - 2$. As $i \in Q$, we conclude that θ can still be improved at \mathbf{x} by a reallocation of one unit of payoff from player k to player i . Thus again $\mathbf{x} \notin v^*(\Gamma_t)$, a contradiction.

(b) Assume $x_i \geq t + 1$. Hence, $N_- \neq \emptyset$ due to $\mathbf{x} \in I^*(\Gamma_t)$. Take any player $k \in N_-$, and let $\epsilon := \max_{k \in S \subseteq N \setminus \{i\}} e(S, \mathbf{x})$. Thereby, $\epsilon \geq e(N_-, \mathbf{x}) \geq -x(N_-)$. Moreover, let $\gamma := \max_{i \in T \subseteq N \setminus \{k\}} e(T, \mathbf{x})$. Since none of the T -type coalitions can receive an aggregate payoff smaller than $x(N_- \setminus \{k\}) + x_i$, we have $\gamma \leq t - (x(N_-) - x_k) - x_i$. So, $\epsilon - \gamma \geq x_i - t - x_k \geq 1 - x_k \geq 2$. Therefore, a reallocation at \mathbf{x} of one unit of payoff from player i to player k improves θ , which means that $\mathbf{x} \notin v^*(\Gamma_t)$, also a contradiction.

In all, it cannot be true that $\mathbf{x} \in v^*(\Gamma_t)$ and $x_i < -1$ or $x_i > t$ for some $i \in N$. \square

Lemma 7. Let Γ_t be a generalized simple game. If $\mathbf{x} \in v^*(\Gamma_t)$, then $-t \leq x_i \leq t$ for all $i \in N$.

Proof. Take any $\mathbf{x} \in v^*(\Gamma_t)$. Note that part (b) of the proof of Lemma 6 does not depend on the monotonicity of Γ_t or on the convention that $v(\emptyset) = 0$. With $I^*(\Gamma_t)$ being redefined as denoting the set $\{\mathbf{x} \in \mathbb{Z}^n : x(N) = t\}$, the assumption that $v(N) = t$ is also not needed. Hence, $x_i \leq t$. Now assume $x_i \leq -(t + 1)$. It then follows from $\mathbf{x} \in I^*(\Gamma_t)$ that there must be a player $k \in N$ with $x_k \geq 1$. Letting $\mu := \max_{i \in S \subseteq N \setminus \{k\}} e(S, \mathbf{x})$ and $\tau := \max_{k \in T \subseteq N \setminus \{i\}} e(T, \mathbf{x})$, we conclude—analogue to part (b) of the proof of Lemma 6—that $\mu - \tau \geq x_k - t - x_i \geq 1 + x_k \geq 2$. Consequently, a reallocation at \mathbf{x} of one unit of payoff from player k to player i improves θ , which contradicts the assumption that $\mathbf{x} \in v^*(\Gamma_t)$. Therefore, if $\mathbf{x} \in v^*(\Gamma_t)$, then $x_i \geq -t$. \square

The following statement can be proved with methods similar to those used in Section 2 (cf. the proofs of Lemmas 1 and 2). The proof is thus omitted:

Lemma 8. Let Γ be a game, and suppose $\mathbf{x} \in v^*(\Gamma)$. If $i \succ_D j$ for two players $i, j \in N$, then $x_i \geq x_j$. If $i \sim_D j$, then $|x_i - x_j| \in \{0, 1\}$.

Remark 8. Let Γ be a game. If \mathbf{z} is a payoff vector that differs from $\mathbf{x} \in v^*(\Gamma)$ only in that the payoffs of two symmetric players are permuted, then $\mathbf{z} \in v^*(\Gamma)$, analogue to Remark 1.

We end our discussion of the integer prenucleolus with a few examples and clarifying remarks:

Example 5. Let Γ_t be represented by $[4; 2, 2, 2, 1; 4]$. In this game, $\mathbf{c} = (6, 6, 6, 4)$. Its integer prenucleolus is composed of all integer preimputations—except for $(1, 1, 1, 1)$ —that satisfy Lemmas 6 and 8: $v^*(\Gamma_t) = \{(2, 1, 1, 0), (1, 2, 1, 0), (1, 1, 2, 0), (2, 2, 1, -1), (2, 1, 2, -1), (1, 2, 2, -1)\}$. Note that player 4 is a null player.

Example 6. Let Γ_t be represented by $[7; 3, 2, 2, 2, 1; 7]$. This is a game without null players, and $\mathbf{c} = (8, 7, 7, 7, 5)$. We get $v^*(\Gamma_t) = \{(2, 2, 2, 2, -1)\}$.

Example 7. Let Γ_t be the directed generalized simple game where $(n, t) = (5, 2)$ and $\mathcal{W}(\Gamma_t) = \{\{1\}, \{2\}, \{3\}, \{4\}, \{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{3, 4\}, \{1, 2, 3\}, \{1, 2, 4\}, \{1, 2, 5\}\}$. In this game, $1 \sim_D 2 \succ_D 3 \sim_D 4 \succ_D 5$. At the same time, e.g., $v(\{1\}) > v(\{1, 5\})$, and hence the game is not monotonic. We find that $\mathbf{c} = (7, 7, 5, 5, 1)$ and $v^*(\Gamma_t) = \{(1, 1, 1, 1, -2)\}$.

Remark 9. Observe for the game in Example 5 that $v^*(\Gamma_t)$ has two elements that weakly preserve \succsim_D . Hence, Lemma 3 and Corollary 1 do not carry over to the integer prenucleolus.

Remark 10. Note for the game in Example 6 that $v(\Gamma_t) = \{(2, 2, 2, 1, 0), (2, 2, 1, 2, 0), (2, 1, 2, 2, 0)\}$. Thus, in contrast to the continuous case, $v^*(\Gamma_t) \neq v(\Gamma_t)$ although $\{i\} \notin \mathcal{W}(\Gamma_t)$ for all $i \in N$.

If negative payoffs are infeasible for a simple game Γ_t with $I(\Gamma_t) = \emptyset$, we may resort to the integer pseudo-nucleolus $v'(\Gamma_t) := \mathcal{N}(\Gamma_t, I'(\Gamma_t))$ as an auxiliary solution concept, where $I'(\Gamma_t) := \{\mathbf{x} \in \mathbb{Z}_{\geq 0}^n : x(N) = v(N)\}$ and $v(N) = t \in \mathbb{Z}_{>0}$. The set $v'(\Gamma_t)$ is nonempty and finite. We leave it to the reader to reconsider Case 2 in the proof of Lemma 3 in order to show that the results of Section 2 carry over to $v'(\Gamma_t)$, noting that the players in the set $F(\Gamma_t) := \{i \in N : \{i\} \in \mathcal{W}(\Gamma_t)\}$ are symmetric and that $i \succ_D j$ for all $i \in F(\Gamma_t), j \in N \setminus F(\Gamma_t)$, due to the monotonicity of Γ_t . We also leave it to the reader to extend the algorithm in Section 3 to the computation of $v^*(\Gamma_t)$ for directed simple games Γ_t . An extended algorithm will admit negative payoffs of some of the players and allow for multiple optimal payoff vectors that weakly preserve \succsim_D .

6. Conclusions

We have been concerned with the class of directed simple games, under the assumption that only integer solutions are feasible. The players of such a game face an integer allocation problem, in that they share a resource that comes in discrete units. We showed as our main theoretical result that the integer nucleolus—if nonempty—of a directed simple game is composed of the images of a particular payoff vector under all symmetries of the game. The respective payoff vector is an integer imputation that weakly preserves the desirability relation between the players. In view of this result, we proposed an algorithm for finding the integer nucleolus of any directed simple game with a nonempty integer imputation set. The algorithm does not need to take account of coalitions that are not winning, and it provides support for computing machinery that can execute multiple threads in parallel. The integer nucleolus of a directed generalized simple game with a nonempty set of integer imputations can be computed in almost the same way. Test runs across samples of weighted voting games on two standard personal computers could be completed within acceptable average time spans. Finding the integer prenucleolus of a directed (generalized) simple game may be more demanding.

We close with a note on the well-known 12-player weighted majority game in Isbell [25] (p. 27). This game has two minimum-sum integer representations, where the quota is 99 and the weight vectors are $\mathbf{w}_1 = (38, 31, 31, 28, 23, 12, 11, 8, 6, 5, 3, 1)$ and, respectively, $\mathbf{w}_2 = (37, 31, 31, 28, 23, 12, 11, 8, 7, 5, 3, 1)$. The weight vectors differ with respect to two non-symmetric players (i.e., players 1 and 9). Assume for this game that t amounts to 197, which is the game's sum of weights. The resulting directed simple game Γ_t has the two representations $[99; \mathbf{w}_1; 197]$ and $[99; \mathbf{w}_2; 197]$. We obtain $\mathbf{c} = (1478, 1388, 1388, 1350, 1304, 1136, 1132, 1104, 1090, 1072, 1056, 1034)$, and $v(\Gamma_t) = \{\mathbf{w}_1\}$.

Acknowledgments: The author is indebted to J. Derks, University of Maastricht, for very helpful comments on earlier versions of this article. Thanks are, moreover, due for a stimulating anonymous communication with respect to the content in Section 5. The remarks and suggestions of two anonymous reviewers also led to an improvement of the article and are gratefully acknowledged.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Schmeidler, D. The Nucleolus of a Characteristic Function Game. *SIAM J. Appl. Math.* **1969**, *17*, 1163–1170.
2. Maschler, M. The Bargaining Set, Kernel, and Nucleolus. In *Handbook of Game Theory with Economic Applications*; Aumann, R.J., Hart, S., Eds.; North-Holland: Amsterdam, The Netherlands, 1992; Volume 1; pp. 591–667.
3. Peleg, B. On Weights of Constant-Sum Majority Games. *SIAM J. Appl. Math.* **1968**, *16*, 527–532.
4. Krohn, I.; Sudhölter, P. Directed and Weighted Majority Games. *Math. Methods Oper. Res.* **1995**, *42*, 189–216.
5. Kurz, S.; Napel, S.; Nohn, A. The Nucleolus of Large Majority Games. *Econ. Lett.* **2014**, *123*, 139–143.
6. Wolff, R.; Karagök, Y. Consistent Allocation of Cabinet Seats: The Swiss Magic Formula. *Public Choice* **2012**, *150*, 547–559.
7. Fragnelli, V.; Gastaldi, F. Remarks on the Integer Talmud Solution for Integer Bankruptcy Problems. *TOP* **2016**, doi:10.1007/s11750-016-0426-z.
8. Fragnelli, V.; Gagliardo, S.; Gastaldi, F. Integer Solutions to Bankruptcy Problems with Non-Integer Claims. *TOP* **2014**, *22*, 892–933.
9. Fragnelli, V.; Gagliardo, S.; Gastaldi, F. Bankruptcy Problems with Non-Integer Claims: Definition and Characterizations of the ICEA Solution. *TOP* **2016**, *24*, 88–130.
10. Lucas-Estañ, M.C.; Gozalvez, J.; Sanchez-Soriano, J. Bankruptcy-Based Radio Resource Management for Multimedia Mobile Networks. *Trans. Emerg. Telecommun. Technol.* **2012**, *23*, 186–201.
11. Leng, M.; Parlar, M. Analytic Solution for the Nucleolus of a Three-Player Cooperative Game. *Naval Res. Logist.* **2010**, *57*, 667–672.
12. Guajardo, M.; Jörnsten, K. Common Mistakes in Computing the Nucleolus. *Eur. J. Oper. Res.* **2015**, *241*, 931–935.
13. Nguyen, T.-D.; Thomas, L. Finding the Nucleoli of Large Cooperative Games. *Eur. J. Oper. Res.* **2016**, *248*, 1078–1092.
14. Taylor, A.; Zwicker, W. *Simple Games: Desirability Relations, Trading, Pseudoweightings*; Princeton University Press: Princeton, NJ, USA, 1999.
15. Maschler, M.; Peleg, B. A Characterization, Existence Proof and Dimension Bounds for the Kernel of a Game. *Pac. J. Math.* **1966**, *18*, 289–328.
16. Aziz, H. Complexity of Comparison of Influence of Players in Simple Games. In *Proceedings of the 2nd International Workshop on Computational Social Choice (COMSOC-2008)*; Endriss, U., Goldberg, P.W., Eds.; Department of Computer Science, University of Liverpool: Liverpool, UK, 2008; pp. 61–72.
17. Aziz, H. Algorithmic and Complexity Aspects of Simple Coalitional Games. Ph.D. Thesis, University of Warwick, Warwick, UK, 2009.
18. Lapidot, E. On Symmetry-Groups of Games. In *Proceedings of the 3rd Annual Israel Conference on Operations Research, Developments in Operations Research*; Avi-Itzhak, B., Passy, U., Joffe, R., Eds.; Gordon and Breach: New York, NY, USA, 1971; Volume 2, pp. 571–583.
19. Lapidot, E. The Counting Vector of a Simple Game. *Proc. Am. Math. Soc.* **1972**, *31*, 228–231.
20. Zoghbi, A.; Stojmenović, I. Fast Algorithms for Generating Integer Partitions. *Int. J. Comput. Math.* **1998**, *70*, 319–332.
21. Hopley, J. Algorithm 152: Nexcom. *Commun. ACM* **1963**, *6*, 68.
22. Carreras, F.; Freixas, J. Complete Simple Games. *Math. Soc. Sci.* **1996**, *32*, 139–155.
23. Hall, M., Jr. *Combinatorial Theory*, 2nd ed.; Wiley: New York, NY, USA, 1986.
24. Gurobi Optimization, Inc. *Gurobi(TM) Optimizer, Version 6.5.0*; Gurobi Optimization, Inc.: Houston, TX, USA, 2015.
25. Isbell, J.R. On the Enumeration of Majority Games. *Math. Tables Other Aids Comput.* **1959**, *13*, 21–28.

