

Document downloaded from:

<http://hdl.handle.net/10251/48387>

This paper must be cited as:

Perea Rojas Marcos, F.; JUSTO PUERTO ALBANDOZ (2013). Finding the Nucleolus of any n-Person Cooperative Game by a Single Linear Program. Computers and Operations Research. 40(10):2308-2313. doi:10.1016/j.cor.2013.03.011.



The final publication is available at

<http://dx.doi.org/10.1016/j.cor.2013.03.011>

Copyright Elsevier

# Finding the Nucleolus of any $n$ -Person Cooperative Game by a Single Linear Program

Justo Puerto<sup>1</sup> and Federico Perea<sup>2</sup>

<sup>1</sup> *IMUS - Universidad de Sevilla (Spain), puerto@us.es*

<sup>2</sup> *Departamento de Estadística e Investigación Operativa Aplicadas y Calidad - Universitat Politècnica de València (Spain), perea@eio.upv.es*

---

## Abstract

In this paper we show a new method for calculating the nucleolus by solving a unique minimization linear program with  $O(4^n)$  constraints whose coefficients belong to  $\{-1, 0, 1\}$ . We discuss the need of having all these constraints and empirically prove that they can be reduced to  $O(k_{max}2^n)$ , where  $k_{max}$  is a positive integer comparable with the number of players. A computational experience shows the applicability of our method over (pseudo)random transferable utility cooperative games with up to 18 players.

*Keywords:* Cooperative games, Nucleolus, Order median problem.

---

## 1. Introduction

This paper presents new advances on computing the nucleolus of a cooperative game with side payments as defined by Schmeidler [19]. Kohlberg [7] proved that the nucleolus can be found by solving a unique linear program of extremely large size with  $O(2^n!)$  constraints. Owen [14] showed how this program can be reduced to a more tractable size of  $O(4^n)$  constraints, although the constraint coefficients are very large. On the other hand, Maschler, Pe-

leg and Shapley [11] gave another method for finding the nucleolus by giving a constructive definition of the lexicographic center of a cooperative game and showing the equivalence between this concept and the nucleolus. In their approach they have to solve a sequence of  $O(4^n)$  minimization linear programs with constraint coefficients in  $\{-1, 0, 1\}$ . This approach was improved by Sankaran [18] who gave a method for computing the nucleolus solving a sequence of only  $O(2^n)$  minimization linear programs with constraint coefficients in  $\{-1, 0, 1\}$ . Hallefjord et al. [6] considered games whose characteristic function is given from mathematical programming problems, and calculated the nucleolus of such games. For this aim, only those coalitions whose characteristic function is required for calculating the nucleolus are considered. Later Potters et al. [17] described a fast algorithm to find the nucleolus of any game with non-empty imputation set. This algorithm is based on solving a prolonged simplex algorithm. It requires solving  $n - 1$  linear programs with at most  $2^n + n - 1$  rows and  $2^n - 1$  columns. Since then, one can find some improvements on the computation of the nucleolus in particular classes of games, but not much has been done on the general case. In this regard, it is worth underlying the paper by Leng and Parlar [10], which develops an algebraic method for finding the nucleolus of any 3-player game with non-empty core. This method is based on a division of different cases depending on the values of the characteristic function.

Despite its computational complexity, the nucleolus has proven very effective in real cost allocation problems. One of the most well-known applications of the nucleolus are the bankruptcy problems, where it was proven by Aumann and Maschler to coincide with the talmudic rule, see [1]. More recent

applications are, for instance: Lemaire [9] applies (among other allocation rules) the nucleolus for allocating costs in insurance companies; Songhuai et al. [20] apply the nucleolus for power losses allocation in electrical markets, which is accepted as it satisfies open, equal, and impartial principles; Le Breton et al. [8] use the nucleolus and the Shapley value for the production and finance of public projects.

In this paper, we show an alternative method for computing the nucleolus of an  $n$ -person cooperative game by solving one unique minimization linear program with  $O(4^n)$  constraints whose coefficients are in  $\{-1, 0, 1\}$ . Although the complexity of the new problem is similar to Owen's one (see [14]), the advantage of the new proposal is that all constraint coefficients are in  $\{-1, 0, 1\}$ , whereas in Owen's formulation some coefficients are extremely large. Besides, we propose a solution method that avoids the problem of having too small constants and significantly reduces the number of constraints. Thus, our formulation gives a computationally more stable method. This special form of the program has proven to be specially suitable for other optimization problems, like the convex order median location problem (see the book by Nickel and Puerto [12]). Recent applications of such problems can be found in Blanco et al. [3] and Espejo et al. [4].

The rest of the paper is structured as follows. Section 2 gives some needed preliminary concepts. In Section 3 we formulate the nucleolus as the solution to a unique LP problem whose resolution is discussed in Section 4. Section 5 illustrates our procedure, which is tested via some experiments in Section 6.

## 2. Background

Cooperative game theory studies decision processes that involve several agents (players) which are allowed to cooperate. This branch of applied mathematics is attracting more and more attention to the scientific community. A proof of that are the ten Nobel prize laureates in Economic Sciences related with game theory (the last two in 2012).

Given the set of players  $N = \{1, \dots, n\}$ , a *coalition* of  $N$  is any subset  $S \subset N$ . The set of all possible coalitions of  $N$  shall be denoted by  $2^N$ . We define the *characteristic function* as the map

$$\begin{aligned} v : 2^N &\longrightarrow \mathbb{R} \\ S &\longrightarrow v(S), \end{aligned} \tag{1}$$

where  $v(S)$  represents the maximum profit that the coalition  $S$  can make by acting on its own, without taking into account what the other players in  $N \setminus S$  can do. So,  $v(N)$  is the best payoff that the coalition formed by all players can obtain. This coalition,  $N$ , is called *the grand coalition*. We set  $v(\emptyset) = 0$ .

Therefore, a cooperative game can be represented by  $\Gamma = (N, v)$  where  $N$  is its set of players  $\{1, 2, \dots, n\}$  and  $v$  is its characteristic function.

One problem faced by cooperative game theory is that of allocating the benefit obtained by the grand coalition among the players. One such allocation is the *nucleolus*, introduced by Schmeidler in [19]. For the definition of the nucleolus, the concept of *excess vector* is needed. The vector of excesses of an allocation  $x \in \mathbb{R}^n$  is the vector  $\theta(x) \in \mathbb{R}^{2^n-1}$  defined as:

$$\theta(x) = (e(S, x)), \quad \text{with } e(S, x) = v(S) - \sum_{i \in S} x_i \quad \forall S \subset N, S \neq \emptyset. \tag{2}$$

The nucleolus is the unique vector that lexicographically minimizes the non-increasing sorted vector of excesses.

Allocations  $x$  such that  $e(S, x) = 0$  if  $S = N$  and  $e(S, x) \leq 0$  for any other coalition  $S$ , are known as core allocations. If the core of a game is nonempty, its nucleolus is a core allocation. Although core allocations have been widely used in the literature for the fairness conditions they satisfy (all coalitions receive at least the value of their characteristic function), there are some recent papers that discuss about some drawbacks of core allocations and go beyond this set (e.g. see Perea et al. [16] or Audy et al. [2].)

In what follows we restrict ourselves to the nucleolus of a cooperative game defined on the set of pre-imputations, namely the set

$$I = \{x \in \mathbb{R}^N : \sum_{i=1}^n x_i = v(N)\}. \quad (3)$$

Note that  $e(N, x) = e(\emptyset, x) = 0$  for all  $x \in I$ , and we therefore consider only  $2^n - 2$  coalitions (all  $S \in 2^N - \{\emptyset, N\}$ .) Needless to say that the results obtained in this paper clearly extend to any polytope.

The following 3-player example illustrates these concepts.

**Example 2.1.** *Consider the following characteristic function:*

$$v(\{i\}) = v(\{1, 3\}) = 0, \quad v(\{1, 2\}) = 3, \quad v(\{2, 3\}) = 1, \quad v(N) = 4,$$

*and the following two pre-imputations:*

$$x = (2.5, 1, 0.5), \quad y = (1.5, 2, 0.5).$$

*Their excess vectors (sorted in a non-increasing way) are:*

$$\theta(x) = (-0.5, -0.5, -0.5, -1, -2.5, -3).$$

$$\theta(y) = (-0.5, -0.5, -1.5, -1.5, -2, -2).$$

Therefore,  $\theta(y) <_L \theta(x)$  (where  $<_L$  means lexicographically smaller). Actually, it can be proven that  $y$  is the nucleolus of this game.

For the sake of readability, we embed the problem in  $\mathbb{R}^{2^n-2} \times \mathbb{R}^n$ , a space of large dimension where the first  $2^n - 2$  coordinates correspond to the excesses (where we assume an ordering on the subsets of  $N$  which is arbitrary but fixed) and the remaining  $n$  to players' allocations. In this space, we deal with the polytope

$$P = \{(\theta, x) \in \mathbb{R}^{2^n-2} \times \mathbb{R}^n : \theta_S \geq v(S) - \sum_{i \in S} x_i, S \subset N \text{ and } \sum_{i=1}^n x_i = v(N)\}.$$

This way we shall simultaneously identify the nucleolus ( $x^*$ ) and its excesses ( $\theta^*$ ).

Let  $(\theta_{(1)}, \dots, \theta_{(2^n-2)})$  be the vector obtained by sorting  $\theta$  in non-increasing order, i.e.,  $\theta_{(1)} \geq \theta_{(2)} \geq \dots \geq \theta_{(2^n-2)}$ . In [13] it is proved that, if  $\lambda_1, \dots, \lambda_{2^n-1}$  are constants sorted in nondecreasing order with  $\lambda_{2^n-1} = 0$ , then  $\sum_{i=1}^{2^n-2} \lambda_i \theta_{(i)}$  can be represented as the solution value of the following linear programming problem:

$$\begin{aligned} \min \quad & \sum_{k=1}^{2^n-2} (\lambda_k - \lambda_{k+1}) (kt_k + \sum_{i=1}^{2^n-2} d_{i,k}) \\ \text{s.t.} \quad & d_{i,k} \geq \theta_i - t_k \quad \forall i, k = 1, \dots, 2^n - 2, \\ & d_{i,k} \geq 0. \end{aligned} \tag{4}$$

In the following section we will prove that, for a convenient choice of the constants  $\lambda$ , a modification of problem (4) gives the nucleolus of any cooperative game.

### 3. The main result

**Theorem 3.1.** *Given a cooperative TU-game  $(N, v)$ , its nucleolus is the unique solution to the following minimization linear program with  $O(4^n)$  variables and constraints whose coefficients are in  $\{-1, 0, 1\}$ :*

$$\begin{aligned}
\min \quad & \sum_{k=1}^{2^n-2} (\lambda_k - \lambda_{k+1}) (kt_k + \sum_{i=1}^{2^n-2} d_{ik}) \\
\text{s.t.} \quad & d_{ik} \geq \theta_i - t_k \quad \forall i, k, \\
& d_{ik} \geq 0 \quad \forall i, k \\
& (\theta, x) \in P,
\end{aligned} \tag{5}$$

with  $\lambda_k = \delta^{k-1}$ ,  $k = 1, \dots, 2^n - 2$ ,  $\lambda_{2^n-1} = 0$  and a convenient choice of  $\delta$ .

**Proof.** The nucleolus  $(\theta^*, x^*)$  corresponds to the lexicographical minimization of the non-increasingly sorted vectors of excesses. Therefore, there exists a permutation  $\sigma(\theta^*)$ , of  $(1, \dots, 2^n - 2)$ , that sorts the elements of the  $\theta$ -variables such that  $(\theta^*, x^*)$  is the lexicographical minimum with respect to the  $\theta$ -variables (excesses).

First of all, it is a folklore result that on compact domains lexicographical minimization is equivalent to linear programming. This can be traced back (at least for finite sets) in [5, p.70] and one explicit proof can be found in the CEnter Discussion Paper No. 20006-89 by S. Tijs. In any case and for the sake of completeness, we prove that after sorting the  $\theta$ -variables according with the permutation  $\sigma(\theta^*)$ , the nucleolus  $(\theta^*, x^*)$  is the unique minimum of the linear function  $(1, \delta, \delta^2, \dots, \delta^{2^n-3}, 0, \dots, 0)(\theta, x)^t$  on  $P_{\sigma(\theta^*)}$ , the polytope that results from  $P$  after reordering the first  $2^n - 2$  coordinates according with the permutation  $\sigma(\theta^*)$ , for some  $\delta < 1$ .



Take  $z \in \text{ext}(P_{\sigma(\theta^*)}) - \{(\theta^*, x^*)\}$ , where  $\text{ext}(Q)$  denotes the set of extreme points of the set  $Q$ . Denote by  $z_{\geq} = (z_{(1)}, \dots, z_{(2^n-2)}, x)$  the vector obtained from  $z$  after sorting its  $2^n - 2$  first components in non-increasing order. Due to the uniqueness of the nucleolus, we know that there exists  $r \in \{1, 2, \dots, 2^n - 2\}$  such that  $\theta_k^* = z_{(k)}$  for  $k < r$  and  $\theta_r^* < z_{(r)}$ . Thus, for any  $\delta > 0$  we have that

$$\begin{aligned}
& (1, \delta, \delta^2, \dots, \delta^{2^n-3}, 0, \dots, 0)[z_{\geq} - (\theta^*, x^*)^t] \\
&= \delta^{r-1}(z_{(r)} - \theta_r^*) + \sum_{k=r+1}^{2^n-2} \delta^{k-1}(z_k - \theta_k^*) \\
&= \delta^{r-1}[(z_{(r)} - \theta_r^*) + \sum_{k=r+1}^{2^n-2} \delta^{k-r}(z_k - \theta_k^*)] \\
&= \delta^{r-1}\mathcal{K}(\delta).
\end{aligned} \tag{6}$$

Note that  $\lim_{\delta \rightarrow 0} \mathcal{K}(\delta) = (z_{(r)} - \theta_r^*) > 0$ . Therefore, there exists  $\delta^*$  so that  $\mathcal{K}(\delta) > 0 \forall \delta < \delta^*$ . Hence, the scalar product in (6) is positive, at least, for any  $\delta < \delta^*$ .

Hence, for all  $\delta < \delta^*$  one has that

$$(1, \delta, \delta^2, \dots, \delta^{2^n-3}, 0, \dots, 0)[z_{\geq} - (\theta^*, x^*)^t] > 0 \forall z \in \text{ext}(P_{\sigma(\theta^*)}) \setminus \{(\theta^*, x^*)\}. \tag{7}$$

However, for  $z = (\theta^*, x^*)^t$ , it attains the null value. Thus,

$$\begin{aligned}
(\theta^*, x^*) &= \arg \min \{(1, \delta, \delta^2, \dots, \delta^{2^n-3}, 0, \dots, 0)(\theta, x)^t : (\theta, x) \in \text{ext}(P_{\sigma(\theta^*)})\} \\
&\quad \forall \delta < \delta^*.
\end{aligned} \tag{8}$$

Thus, once the permutation that gives the lexicographic ordering in the optimum is known, finding the nucleolus reduces to solving a linear program. Nevertheless, in order to apply the above argument we need to prove that

the problem that gives the lex-minimum, namely

$$\begin{aligned}
\min \quad & \sum_{i=1}^{2^n-2} \delta^{i-1} \theta_{(i)} \\
\text{s.t.} \quad & \theta_{(1)} \geq \theta_{(2)} \geq \dots \geq \theta_{(2^n-3)} \\
& (\theta, x) \in P
\end{aligned} \tag{9}$$

can be written as a linear programming problem. This formulation is doable using the result in [13], see problem (4). Consider linear programming problem (5). The objective function and the first group of constraints represent the ordered weighted sum of the values  $\sum_{k=1}^{2^n-2} \lambda^k \theta_{(k)}$ , where  $\theta_{(1)} \geq \theta_{(2)} \geq \dots \geq \theta_{(2^n-2)}$ . Notice that this formulation results from the reformulation of the ordered median problem that appears in [13, Section 3] (see also [12]). It is clearly applicable here because we consider the convex case of the weighted ordered average, i.e  $\delta^0 \geq \delta^1 \geq \dots \geq \delta^{2^n-3} \geq 0$ . This formulation, together with the fact that for the permutation  $\sigma(\theta^*)$ ,  $(\theta^*, x^*)$  is the unique minimum of  $(1, \delta, \delta^2, \dots, \delta^{2^n-3}, 0 \dots 0)(\theta, x)^t$  on  $P_{\sigma(\theta^*)}$ , proves that computing the nucleolus of an  $n$ -person cooperative game is equivalent to solving the continuous linear program (5), which has  $O(4^n)$  variables,  $O(4^n)$  constraints with coefficients in  $\{-1, 0, 1\}$ .

The uniqueness in the solution to (5) comes from the fact that, on compact and convex domains, the nucleolus is unique (see for instance [19] or [7]). Because we are dealing with allocations in the compact and convex set  $I$ , see (3), this uniqueness result follows.  $\square$

The reader may have noted that a key aspect in the previous proof is the calculus of  $\delta^*$ . The following lemma gives an explicit estimate of an upper bound for such a constant.

**Lemma 3.1.** *Let  $\beta = \min\{(z_{(r)} - z'_{(r)}) : z, z' \in \text{ext}(P), r \in \{1, \dots, 2^n -$*

2} so that  $z_{(k)} = z'_{(k)} \forall k < r$ ,  $z_{(r)} > z'_{(r)}$ , and let  $\delta^* = \frac{\beta}{2v(N)+\beta}$ . Then, for all  $\delta < \delta^*$  we have that  $\mathcal{K}(\delta) > 0$ .

**Proof.** Let  $\theta^*$  be the excesses produced by the nucleolus  $x^*$  sorted in a non-increasing way. Take  $z \in \text{ext}(P) \setminus \{\theta^*, x^*\}$  in the conditions of the lemma, and let  $r \in \{1, 2, \dots, 2^n - 2\}$  be such that  $\theta_k^* = z_{(k)}$  for  $k < r$  and  $\theta_r^* < z_{(r)}$ . It is clear that  $(z_{(r)} - \theta_r^*) \geq \beta$ . Thus, if  $\delta < \delta^*$ , we also have that  $\delta < \frac{z_{(r)} - \theta_r^*}{2v(N) + (z_{(r)} - \theta_r^*)}$ , because function  $f(y) = y/(a + y)$  is monotone increasing for all  $y \neq a$  if  $a > 0$ . Therefore the following inequalities hold:

$$\delta < \frac{(z_{(r)} - \theta_r^*)/2v(N)}{1 + (z_{(r)} - \theta_r^*)/2v(N)} \Rightarrow \delta + \delta \frac{(z_{(r)} - \theta_r^*)}{2v(N)} < \frac{(z_{(r)} - \theta_r^*)}{2v(N)},$$

which implies that

$$\frac{\delta}{1 - \delta} < \frac{(z_{(r)} - \theta_r^*)}{2v(N)}.$$

Since  $\delta < \delta^* < 1$ , the last inequality implies

$$\sum_{k=2}^{\infty} \delta^{k-1} < \frac{(z_{(r)} - \theta_r^*)}{2v(N)} \Rightarrow \sum_{k=r+1}^{2^n-2} \delta^{k-1} < \frac{(z_{(r)} - \theta_r^*)}{2v(N)} \Rightarrow$$

$$\sum_{k=r+1}^{2^n-2} \delta^{k-1} 2v(N) < (z_{(r)} - \theta_r^*) \Rightarrow \sum_{k=r+1}^{2^n-2} \delta^{k-1} (v(N) - (-v(N))) < (z_{(r)} - \theta_r^*).$$

Now, since the excesses are within  $[-v(N), v(N)]$ , it follows that  $\theta_k^* < v(N)$  and  $z_{(k)} > -v(N)$ . Therefore

$$\sum_{k=r+1}^{2^n-2} \delta^{k-1} (\theta_k^* - z_{(k)}) < (z_{(r)} - \theta_r^*) \Rightarrow (z_{(r)} - \theta_r^*) + \sum_{k=r+1}^{2^n-2} \delta^{k-1} (z_{(k)} - \theta_k^*) > 0.$$

□

We note that calculating this upper bound may be a difficult task depending on the structure of the polytope  $P$ . In our computational experiments we have taken  $\delta = 0.1$  with satisfactory results.

#### 4. Computational aspects

We have proven in Theorem 3.1 that the nucleolus can be solved by means of the following LP problem:

$$\begin{aligned}
\min \quad & \sum_{k=1}^{2^n-2} (\lambda_k - \lambda_{k+1})(kt_k + \sum_{i=1}^{2^n-2} d_{i,k}) \\
\text{s.t.} \quad & d_{i,k} \geq \theta_i - t_k \quad \forall i, k = 1, \dots, 2^n - 2, \\
& \theta_i = v(S_i) - \sum_{j \in S_i} x_j, \quad \forall i = 1, \dots, 2^n - 2, \\
& \sum_{j=1}^n x_j = v(N) \\
& d_{i,k} \geq 0
\end{aligned} \tag{10}$$

with  $\lambda_k = \delta^{k-1}$ ,  $k = 1, \dots, 2^n - 2$  and  $\lambda_{2^n-1} = 0$ .

Although our approach proves that the nucleolus is computable by this single LP problem, this method may be affected by current implementation of LP solvers due to actual precision in representing primitives (scalars).

A problem of this formulation is that, due to the small magnitude of constants  $\delta^k$ , they may be considered as zero by computers when  $k$  is “large”. In order to solve this drawback, we propose the following iterative process.

Assume the maximum  $k$  so that  $\delta^k$  can be ensured strictly positive by computer precision is  $k_{max} - 1$ . Therefore, we set  $\lambda_k = 0 \quad \forall k > k_{max}$ . With these new values, the objective function in (10) becomes

$$\sum_{k=1}^{k_{max}} (\lambda_k - \lambda_{k+1})(kt_k + \sum_{i=1}^{2^n-2} d_{i,k}). \tag{11}$$

Note that, after this adaptation, variables  $t_k, d_{ik}$  do not appear in this objective function for all  $k > k_{max}$ , and therefore the first set of constraints of problem (10) can be reduced to

$$d_{i,k} \geq \theta_i - t_k \quad \forall i = 1, \dots, 2^n - 2, k = 1, \dots, k_{max}, \tag{12}$$

which consists of  $O(k_{max}2^n)$  constraints instead of  $O(4^n)$ . The reader may observe that the solution to this problem lexicographically minimizes the  $k_{max}$  largest excesses. If the solution to this problem gives the nucleolus, we stop. Otherwise, we let  $T_k = (kt_k^1 + \sum_{i=1}^{2^n-2} d_{i,k}^1)$  for  $k = 1, \dots, k_{max}$ , where  $t^1$  and  $d^1$  are the values of variables  $t$  and  $d$  in the optimal solution.  $T_k$  is the sum of the  $k^{th}$  largest excesses (see [13], Lemma 1). If this problem does not yield the nucleolus, in a following iteration we fix the values of the  $k_{max}$  largest excesses to be equal to those calculated before, and lexicographically minimize the  $k_{max} + 1, \dots, 2k_{max}$  largest excesses.

Therefore the objective function becomes

$$\sum_{k=k_{max}+1}^{2k_{max}} (\lambda_{k-k_{max}} - \lambda_{k+1-k_{max}})(kt_k + \sum_{i=1}^{2^n-2} d_{i,k}). \quad (13)$$

The first set of constraints reduces to

$$d_{i,k} \geq \theta_i - t_k \quad \forall i = 1, \dots, 2^n - 2, k = 1, \dots, 2k_{max}. \quad (14)$$

And we now add this new set of constraints:

$$kt_k + \sum_{i=1}^{2^n-2} d_{i,k} = T_k, k = 1, \dots, k_{max}, \quad (15)$$

which aim at fixing the  $k_{max}$  largest excesses. If this procedure does not give the nucleolus, we store  $T_k = (kt_k^2 + \sum_{i=1}^{2^n-2} d_{i,k}^2)$  for  $k = k_{max} + 1, \dots, 2k_{max}$ , where  $t^2$  and  $d^2$  are the optimal values of variables  $t$  and  $d$  in this second problem.

Therefore, in this iterative process, the  $m$ -th iteration solves the following

linear programming problem:

$$\begin{aligned}
\min \quad & \sum_{k=(m-1)k_{max}+1}^{mk_{max}} (\lambda_{k-(m-1)k_{max}} - \lambda_{k+1-(m-1)k_{max}}) (kt_k + \sum_{i=1}^{2^n-2} d_{i,k}) \\
\text{s.t.} \quad & d_{i,k} \geq \theta_i - t_k \quad \forall i = 1, \dots, 2^n - 2, k = 1, \dots, mk_{max}, \\
& \theta_i = v(S_i) - \sum_{j \in S_i} x_j, \quad \forall i = 1, \dots, 2^n - 2, \\
& \sum_{j=1}^n x_j = v(N), \\
& (kt_k + \sum_{i=1}^{2^n-2} d_{i,k}) = T_k, \quad k = 1, \dots, (m-1)k_{max}, \\
& d_{i,k} \geq 0.
\end{aligned} \tag{16}$$

Note that problem (16) lexicographically minimizes the  $(m-1)k_{max}+1, \dots, mk_{max}$  largest excesses, while the first  $(m-1)k_{max}$  are fixed to the values found in previous iterations. This way we guarantee that the  $mk_{max}$  largest excesses are lexicographically minimized.

This process should stop when, among the excesses that have been lexicographically minimized, there is a value that is unique, see [15] page 331–332. The solution to the last LP problem would give the nucleolus of the game. A pseudocode of this process is given in Algorithm 1.

We would like to emphasize that the above iterative approach is not theoretically necessary since our approach obtains the nucleolus by solving a unique LP problem. Nevertheless we have tested that in actual computational experiments this algorithm helps to avoid problems with tolerance of current LP solvers and speeds our process up.

**Data:** A TU-game  $(N, v)$

Set  $STOP := False$  and  $m = 1$ ;

**while**  $STOP = False$  **do**

    Solve (16);

**if** *there is a unique value among the  $mK_{max}$  first excesses* **then**

$STOP = True$ ;

**else**

$T_k := kt_k + \sum_{i=1}^{2^n-2} d_{i,k}, \forall k = 1, \dots, mK_{max}$  ;

$m \rightarrow m + 1$  ;

**end**

**end**

**Result:**  $(x^*, \theta^*)$ , the nucleolus of  $(N, v)$  and its excesses.

**Algorithm 1:** Iterative process to solve problem (5).

## 5. Illustrative example

In this section we show the applicability of our approach by calculating the nucleolus of a 14-player game. The ordering of the coalitions  $S_k$ ,  $k = 1, \dots, 2^{14} = 16384$ , is such that  $k = 1 + \sum_{j \in S_k} 2^{j-1}$ , with  $S_1 = \emptyset$ .

The characteristic function of each coalition  $S_k$  is calculated as  $v(S_k) = 0$  if  $\exists j = 1, \dots, n : 2^{j-1} + 1 = k$  or  $k = 1$ ,  $v(S_{2^n}) = 1$ , and  $v(S_k) = \frac{1}{(n(n+1)/2)} \sum_{j \in S_k} (j - \text{mod}(k, j))$  otherwise. Function  $\text{mod}(k, j)$  yields the remainder when dividing  $k$  by  $j$ . Adding this function in the definition of the characteristic function gives a pseudorandom aspect to these games, although they can be replicated easily. Note that  $v(\emptyset) = v(\{j\}) = 0 \forall j$  and  $v(N) = 1$ . Note as well that  $v(S) \in (0, 1)$  for any other coalition  $S$ .

### 5.1. Solution in one step

Our procedure, taking  $k_{max} = 20$  and  $\delta = 0.1$ , obtained the following allocation:

$$\begin{aligned} x_1 &= 0.032074, x_2 = 0.017313, x_3 = 0.038355, x_4 = 0.032074, x_5 = 0.054903, \\ x_6 &= 0.054392, x_7 = 0.059653, x_8 = 0.076455, x_9 = 0.085445, x_{10} = 0.091726, \\ x_{11} &= 0.098008, x_{12} = 0.106997, x_{13} = 0.123545, x_{14} = 0.129061. \end{aligned}$$

The largest excess is  $-0.009500$ . We emphasize that the 16<sup>th</sup> excess is the unique excess whose value is  $-0.010776$ , meaning that both the previous and the following excesses are different. Therefore, there is no other allocation  $x'$  that yields the same excesses. Because of that we can confirm that the given allocation is the nucleolus. If we did not have such unique value within the first 20 excesses, we would have needed to solve the problem again calculating any other number of excesses strictly greater than 20 (as detailed in Algorithm 1, where we calculated the first  $2 \times 20$  excesses). The algorithm would stop when at least one of the first largest excesses yields a unique value.

The running time was 34 seconds. This example was solved on a Packard Bell computer, with Intel(R) Core(TM) i7 2.80 GHz processor, 6 GB of RAM memory, running on Windows 7 (64 bits). The rest of experiments were solved in the same computer.

### 5.2. Solution in two steps

It may happen that the allocation obtained is not guaranteed to be the nucleolus, because none of the  $k_{max}$  largest excesses yields a unique value. For instance, consider  $k_{max} = 10$ . Our iterative process would run as follows:



1.  $m = 1$ . The solution obtained is the same as the one before. The difference is that now we can only guarantee that the 10 largest excesses are lexicographically minimum. Note that among the 10 largest excesses (which are the same as before) there is not a unique value, and therefore we cannot guarantee that the obtained solution is the nucleolus. This first iteration took 31 seconds. In this case, we would proceed to the next iteration, imposing the values  $T_k$  for the 10 largest excesses.
2.  $m = 2$ . We now solve the problem for this value of  $m$ . The new solution is the same as before. Now, since we can guarantee that the first 20 excesses are lexicographically minimum, we can therefore guarantee that the solution obtained is the nucleolus, and we stop the process. This second iteration took 79 seconds.

## 6. Computational results

In this section we show some characteristics observed when calculating the nucleoli of the family of games defined in Section 5 for number of players  $n = 10, \dots, 18$ . Table 1 shows the obtained results for the different values of  $n$ . More specifically, the columns refer to the  $k_{max}$  used, the first sorted excess whose value is unique, the constant  $\delta$  used, and the computational time needed to solve the corresponding problems. The nucleoli for these games are shown in the appendix. All our codes are written in GAMS 23.0.6 using CPLEX 11.2.1 and are available upon request.

We first note that the constant  $\delta$  could be fixed to 0.1 in all cases. We also noted that the first unique value was always among the 20 largest excesses, except for  $n = 18$ , although we can observe an increase in such value with  $n$ .

$n$	$k_{max}$	First unique	$\delta$	Time	Iter
10	20	13	0.1	0.62	1
11	20	13	0.1	1.06	1
12	20	14	0.1	2.80	1
13	20	16	0.1	10.44	1
14	20	16	0.1	34.80	1
15	20	16	0.1	117.10	1
16	20	20	0.1	432.93	1
17	20	19	0.1	1666.70	1
18	10	20	0.1	6490.00 + 6456.41 + 13054.40	3

Table 1: Results. Columns mean: number of players, maximum power of  $\delta$ , first unique excess, value of  $\delta$ , computation time (in seconds), and number of iterations needed with the considered parameters. See the obtained nucleoli for each game in the Appendix.

As a matter of fact, fitting these data to the linear function

$$k^* = 0.9881n + 2.5357, \quad (17)$$

where  $k^*$  denotes the first excess that yields a unique value, has a linear coefficient of determination  $R^2 = 0.8748$ , and we therefore could consider a linear increase in this value with respect to  $n$ . Finally, we note that when fitting the computational time vs.  $n$  via the exponential function

$$Time = 3 \times 10^{-6} e^{1.1636n}, \quad (18)$$

the coefficient of determination is  $R^2 = 0.9915$ . Therefore assuming a (logical) exponential increase in computational time with  $n$  seems reasonable.

Finally, for  $n = 18$  we considered  $k_{max} = 10$  in order to reduce the number of constraints, due to memory problems in the computer used. Note that in this case we needed three iterations.

## 7. Conclusions

In this paper we have presented a procedure to calculate the nucleolus of any benefit game, without restrictions of non-emptiness in imputation set, core set, etc. The approach consists of solving a unique linear programming problem. Since the formulation of this problem involves a constant smaller than one powered to large exponents, one could find some troubles when solving it because some of these powers might be considered zero by computers' precision. In order to avoid this problem, we have proposed an iterative method that stops when the obtained solution is guaranteed to be the nucleolus. Such condition is met when, among the largest excesses that have been lexicographically minimized, at least one of them is unique.

In order to illustrate our method, we have detailed the calculation of a 14-player game's nucleolus. The applicability of our method has been shown by calculating the nucleoli of games with number of players ranging from 10 to 18. The characteristic function of such games is deterministic and has been given implicitly, and therefore these experiments could be replicated by other researches working on the nucleolus or other allocations for TU games.

### **Acknowledgments**

The authors want to thank Javier Arin, Safae El Haj Ben Ali, Guillermo Owen and Johannes H. Reijnierse for their useful and valuable help. The research of the authors has been partially supported by the projects FQM-5849 (Junta de Andalucía\FEDER), and by the project MTM2010-19576-C02-01 (MICINN, Spain). This paper was written while the second author was enjoying a grant for a short postdoctoral research visit at the *Instituto Universitario de Investigación de Matemáticas de la Universidad de Sevilla (IMUS)*. Special thanks are due to one anonymous referee for his/her valuable comments.

### **Appendix: Obtained nucleoli**

See Table 2.

### **References**

- [1] Aumann, R., Maschler, M., 1985. Game theoretic analysis of a bankruptcy problem from the Talmud, *J. Econ. Theor.* 36, 195–213.

$n$	$x \in \mathbb{R}^n$
10	(0.063633, 0.021211, 0.081767, 0.063667, 0.103044, 0.099967, 0.093978, 0.139344, 0.157644, 0.175744)
11	(0.059333, 0.017050, 0.054167, 0.054267, 0.083983, 0.077067, 0.082717, 0.114867, 0.144683, 0.144583, 0.167283)
12	(0.046421, 0.018047, 0.049729, 0.047156, 0.070918, 0.066406, 0.070182, 0.098456, 0.119544, 0.122118, 0.141368, 0.149656)
13	(0.041100, 0.012400, 0.041960, 0.040980, 0.059100, 0.057480, 0.061900, 0.084880, 0.102020, 0.103000, 0.119400, 0.128780, 0.147000)
14	(0.032074, 0.017313, 0.038355, 0.032074, 0.054903, 0.054392, 0.059653, 0.076455, 0.085445, 0.091726, 0.098008, 0.106997, 0.123545, 0.129061)
15	(0.031320, 0.015888, 0.033756, 0.030507, 0.044315, 0.048554, 0.051268, 0.067690, 0.074366, 0.077615, 0.084568, 0.097107, 0.107132, 0.114163, 0.121751)
16	(0.031786, 0.011048, 0.031124, 0.025826, 0.040679, 0.040752, 0.043212, 0.058217, 0.064781, 0.070079, 0.077252, 0.084626, 0.096588, 0.101050, 0.108729, 0.114252)
17	(0.020468, 0.014736, 0.027505, 0.024377, 0.034654, 0.041935, 0.039837, 0.053386, 0.057626, 0.061073, 0.067354, 0.076677, 0.084045, 0.088590, 0.096364, 0.102777, 0.108596)
18	(0.022618, 0.011218, 0.026209, 0.022577, 0.029782, 0.035802, 0.034739, 0.051877, 0.049550, 0.058861, 0.058982, 0.069377, 0.070682, 0.079432, 0.084850, 0.092777, 0.098457, 0.102209)

Table 2: For each value of  $n$  (number of players), the nucleolus of the corresponding pseudo-random game we have designed is given.

- [2] Audy, J., D'Amours, S., Rönnqvist, M., 2012. An empirical study on coalition formation and cost/savings allocation. *International Journal of Production Economics* 136 (1), 13–27.
- [3] Blanco, V., El-Haj Ben-Ali, S., Puerto, J., 2012. Minimizing ordered weighted averaging of rational functions with applications to continuous location. *Computers and Operations Research*. In Press.
- [4] Espejo, I., Rodríguez-Chía, A. M., Valero, C., 2009. Convex ordered median problem with lp-norms. *Computers and Operations Research* 36 (7), 2250–2262.
- [5] Fishburn, P., 1970. *Utility Theory for Decision Making*. Wiley, New York.
- [6] Hallefjord, A., Helming, R., Jornsten, K., 1995. Computing the nucleolus when the characteristic function is given implicitly: a constraint generation approach. *International Journal of Game Theory* 24, 357–372.
- [7] Kohlberg, E., 1972. The nucleolus as a solution of a minimization problem. *SIAM J. Appl. Math.* 23, 34–39.
- [8] Le Breton, M. Moreno-Tertero, J.D., Savvateev, A. Weber, S., 2011. Stability and fairness in models with a multiple membership. *Int J Game Theory* (In press).
- [9] Lemaire, J., 1984. An application of game theory: cost allocation. *Astin Bulletin* 14 (1), 61–81.

- [10] Leng, M., Parlar, M., 2010. Analytic solution for the nucleolus of a three-player cooperative game. *Naval Research Logistics* 57 (7), 667–672.
- [11] Maschler, M., Peleg, B., Shapley, L., 1979. Geometric properties of the kernel, nucleolus and related solution concepts. *Mathematics of Operations Research* 4, 303–338.
- [12] Nickel, S., Puerto, J., 2005. *Location theory — a unified approach*. Springer.
- [13] Ogryczak, W., Tamir, A., 2003. Minimizing the sum of the  $k$  largest functions in linear time. *Information Processing Letters* 85, 117–122.
- [14] Owen, G., 1974. A note on the nucleolus. *International Journal of Game Theory* 3, 101–103.
- [15] Owen, G., 1995. *Game theory*, 3rd Edition. Academic Press, Inc., London.
- [16] Perea, F., Puerto, J., Fernández, F. R., 2012. Avoiding unfairness of owen allocations in linear production processes. *European Journal of Operational Research* 220 (1), 125–131.
- [17] Potters, J., Reijnierse, J., Ansing, M., 1996. Computing the nucleolus by solving a prolonged simplex algorithm. *Math. Operations Research* 21, 757–768.
- [18] Sankaran, J., 1991. On finding the nucleolus of an  $n$ -person cooperative game. *International Journal of Game Theory* 19, 329–338.

- [19] Schmeidler, D., 1969. The nucleolus of a characteristic function game. SIAM J. Appl. Math. 17, 1163–1170.
- [20] Songhuai, D., Xinghua, Z., Lu, M., Hui, X., 2006. A Novel Nucleolus-Based Loss Allocation Method in Bilateral Electricity Markets. IEEE Transactions on Power Systems 21 (1), 28–33.