European Journal of Operational Research 241 (2015) 931–935

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Short Communication

Common mistakes in computing the nucleolus

Mario Guajardo*, Kurt Jörnsten

Department of Business and Management Science, NHH Norwegian School of Economics, N-5045 Bergen, Norway

ARTICLE INFO

Article history:
Received 31 March 2014
Accepted 18 October 2014
Available online 26 October 2014

Keywords:
Game theory
Nucleolus
Cost allocation
Linear programming
Duality

ABSTRACT

Despite linear programming and duality have correctly been incorporated in algorithms to compute the nucleolus, we have found mistakes in how these have been used in a broad range of applications. Overlooking the fact that a linear program can have multiple optimal solutions and neglecting the relevance of duality appear to be crucial sources of mistakes in computing the nucleolus. We discuss these issues and illustrate them in five mistaken examples from this and other journals. The purpose of this note is to prevent these mistakes propagate longer by clarifying how linear programming and duality can be correctly used for computing the nucleolus.

1. Introduction

One of the main solution concepts in cooperative game theory is the nucleolus, proposed by Schmeidler (1969). A number of approaches have been developed in order to compute it, as reviewed by Leng and Parlar (2010) and Çetiner (2013). Although linear programming and duality have been correctly used in several approaches (e.g. Fromen, 1997; Hallefjord, Helming, & Jørnsten, 1995; Kimms & Çetiner, 2012), we have found that the nucleolus has been wrongly computed over the years in a wide variety of contexts. The mistakes appear to be caused by overlooking the possibility that a linear program can have multiple solutions, and by neglecting the use of the dual solution as a valuable source of information in such cases. In this short communication, we discuss these issues and illustrate them in five examples taken from articles published in this journal and others. The examples correspond to applications of cooperative game theory in joint development of projects (Kruś & Bronisz, 2000), production and transportation planning (Sakawa, Nishizaki, & Uemura, 2001), electricity markets (SatyaRamesh & Radhakrishna, 2009), manufacturing (Oh & Shin, 2012), and investments (Lemaire, 1984). It came to our attention that similar errors have appeared in such a wide range of applications. Our purpose in this note is to clarify how linear programming and duality can be used to correctly calculate the nucleolus, thus to prevent an even larger propagation of these errors. The clarification on how to use these concepts is presented in Section 2 of this note. In Section 3, we present the mistaken examples from previous literature and compute the correct nucleolus to them. In Section 4, we conclude with some final remarks.

2. The nucleolus of a cooperative game and linear programming

Let $N = \{1, \dots, n\}$ be the set of players and K the set of all nonempty subsets of N. The characteristic function $v : K \to \mathbb{R}$ assigns to each coalition S in K the *cost* of coalition S. A preimputation or cost allocation vector $x = (x_1, \dots, x_n)$ assigns to each player j in N a quantity x_j such that $\sum_{j \in N} x_j = v(N)$; that is, the cost of the grand coalition N is split among its members according to the allocation x ($x_j \in \mathbb{R}\ \forall j \in N$). An allocation vector x satisfies rationality if $\sum_{j \in S} x_j \leq v(S)\ \forall S \in K$. The core of the game is the set of preimputations that satisfy the rationality conditions.

Define the excess of coalition S at x as $\varepsilon(x, S) = v(S) - \sum_{j \in S} x_j$. The excess is a measure of how *satisfied* a coalition S is with the cost allocation x. The larger the excess of S, the more satisfied coalition S is. Define the excess vector at x as $e(x) = (\varepsilon(x, S_1), \dots, \varepsilon(x, S_m))$, where the sets S_i represent the coalitions in $K \setminus N$, and $m = 2^n - 2$. For an excess vector $e \in \mathbb{R}^m$, define a mapping θ such that $\theta(e) = y$, where $y \in \mathbb{R}^m$ is the vector which results from arranging the components of e in a non-decreasing order. A vector $y = (y_1, \dots, y_m)$ is said to be lexicographically greater than another vector $\bar{y} = (\bar{y}_1, \dots, \bar{y}_m)$ if either $y = \bar{y}$ or there exists $h \in \{1, \dots, m\}$ such that $y_h > \bar{y}_h$ and $y_i = \bar{y}_i\ \forall i < h$ (if $h = 1$, it is enough that $y_h > \bar{y}_h$). We annotate $y \succeq \bar{y}$.

Note in some contexts the characteristic function v is defined as a benefit instead of cost, and the excess as a measure of dissatisfaction instead of satisfaction. Both perspectives can be approached in equivalent ways. We rather adopt the cost perspective, since most of the recent interest for cooperative games in Operations Research comes from cost sharing problems in collaborative logistics. Also, our

* Corresponding author. Tel.: +47 55959834; fax: +47 55959650.
E-mail addresses: mario.guajardo@nhh.no (M. Guajardo), kurt.jornsten@nhh.no (K. Jörnsten).

http://dx.doi.org/10.1016/j.ejor.2014.10.037
0377-2217/

attention focus in games with a non-empty core. A main question in these games is how the players should share the cost $v(N)$ when collaborating in the grand coalition N. The nucleolus is one of the most used solution concepts for this problem.

The nucleolus of a cost sharing game with non-empty core can be defined as the preimputation x which lexicographically maximizes the excess vector, that is, $\theta(e(x)) \succeq \theta(e(\bar{x}))$ for all preimputation $\bar{x}$. Schmeidler (1969) proves that the nucleolus is unique. In order to compute the nucleolus, let us first consider the following linear programming model (henceforth denoted as P), which looks for a preimputation $x = (x_1, \ldots, x_n)$ that maximizes the minimum excess ε among all the coalitions.

$$(P) \qquad \max \ \varepsilon \tag{1}$$

$$\text{s.t.} \quad \varepsilon + \sum_{j \in S} x_j \le v(S) \qquad \forall S \subset N, S \neq \emptyset \tag{2}$$

$$\sum_{j \in N} x_j = v(N) \tag{3}$$

$$\varepsilon \in \mathbb{R}, \ \ x_j \in \mathbb{R} \ \forall j \in N \tag{4}$$

Objective function (1) maximizes ε. Constraints (2) impose that such ε cannot be greater than the excess of any coalition. Thus, (1) and (2) together provide that ε is exactly equal to the minimum excess. Constraint (3) is the efficiency condition, which provides that the cost of the grand coalition $v(N)$ is split among its players according to the allocation x. Constraints (4) state the nature of the variables. The solution to P is not necessarily unique. As we will illustrate in the numerical examples, it may occur that more than one allocation x leads to the optimal objective value. In addition, a solution of P provides an allocation that maximizes the lowest excess, but not necessarily the second or any subsequent lowest excess. The nucleolus can be found by solving a sequence of linear programs (LPs), as in the algorithm by Fromen (1997) which we briefly outline below. The first LP in the sequence corresponds to P. Let ε_1 be the optimal objective value of P. The kth LP ($k > 1$) in the sequence is formulated below.

$$\max \varepsilon_k \tag{5}$$

$$\text{s.t.} \quad \varepsilon_k + \sum_{j \in S} x_j \le v(S) \qquad \forall S \subset N : S \notin \mathcal{F}_k \tag{6}$$

$$\varepsilon_i + \sum_{j \in S} x_j = v(S) \qquad \forall S \in F_i, \ i \in \{1, \ldots, k-1\} \tag{7}$$

$$\sum_{j \in N} x_j = v(N) \tag{8}$$

$$\varepsilon_k \in \mathbb{R}, \ \ x_j \in \mathbb{R} \ \forall j \in N \tag{9}$$

In this kth LP, objective function (5) and constraints (6) provide that the kth minimum excess ε_k is maximized. Constraints (7) state that the excess of the coalitions contained in set F_i must be equal to the optimal objective value ε_i to the ith LP. Constraints (8) and (9) state conditions for the efficiency and nature of the variables, respectively. The set F_i is the set of all coalitions for which the excess constraint (6) is satisfied with equality sign for all the solutions to the ith LP. Thus, the excess of the coalitions in F_i must be fixed to ε_i in the kth LP in the series for all $k > i$, as expressed in constraint (7). The set $\mathcal{F}_k$ is simply the union of all the coalitions for which its excess has been fixed in a previous LP in the sequence, that is, $\mathcal{F}_k = \bigcup_{i<k} F_i$. Note by defining $\mathcal{F}_1 = \emptyset$ and omitting constraints (7) for $k = 1$, one recovers the first problem P in the sequence. A key issue is how to find the set F_i, and here is where dual linear programming plays a relevant role. The dual of P, which we will refer as model D, is formulated below.

$$(D) \qquad \min \sum_{S \in K} v(S) \cdot y_S \tag{10}$$

$$\text{s.t.} \quad \sum_{S \in K \setminus N} y_S = 1 \tag{11}$$

$$\sum_{S \in K : j \in S} y_S = 0 \quad \forall j \in N \tag{12}$$

$$y_S \ge 0 \quad \forall S \in K \setminus N, \quad y_N \in \mathbb{R} \tag{13}$$

From duality theory, when the optimal value of a dual variable is positive, the inequality constraint associated to this variable must hold with equality at any optimal solution of P. Therefore, given a solution to P the set F_1 can be formed by all the coalitions S for which y_S is positive in the corresponding solution to D. Analogously, for a general k, the set F_k can be formed by all the coalitions such that the dual variable associated to constraint (6) is positive in the corresponding optimal solution to the dual problem of the kth LP in the sequence. In order to find the nucleolus, the solution process proceeds until a k where the LP has a unique solution. At the latest, such unique solution will be obtained when constraints (7) and (8) define a system of n independent linear equations.

Strictly speaking, the previous procedure computes the *prenucleolus* of the game. The nucleolus requires x not only to be a preimputation but also to satisfy the individual rationality constraint $x_j \le v(\{j\}) \ \forall j \in N$. This can be explicitly added as a constraint in the LPs for games whose core may be empty. However, in games with non-empty core the prenucleolus coincides with the nucleolus, so the explicit inclusion of this constraint is not needed. Also, notice that for games with non-empty core the optimal objective value is non-negative for all the LPs in the sequence, thus one can declare $\varepsilon_k \ge 0$ instead of $\varepsilon_k \in \mathbb{R}$.

3. Numerical examples

In this section we present five examples taken from a variety of contexts in the literature, where the nucleolus has been wrongly calculated. The first two examples are taken from articles published in this journal and the other three examples from other journals. We identify two main sources of error. First, overlooking the fact that the solution to model P is not unique. Second, given a particular solution to the ith LP in the sequence, the set F_i has been wrongly computed as the set of all coalitions whose excess is equal to ε_i at such particular solution.

We use the notation $\hat{v}(S)$ for referring to the characteristic function of games where the players share benefits instead of costs (the LP models for these games remain the same as in Section 2 by defining $v(S) = -\hat{v}(S)$).

3.1. Joint projects

Kruś and Bronisz (2000) consider a cooperative game where different agents are interested in the implementation of a project. The authors outline a correct algorithm for calculating the nucleolus (and other nucleoli variants) based on a sequence of LPs. They correctly acknowledge that the solution to an LP in the sequence may not have a unique solution, and also that the optimal dual solution is useful for the implementation of the algorithm. They refer the reader to Christensen, Lind, and Tind (1996), who indeed incorporate the information of the dual values in the solution process correctly. Despite the correctness of the algorithm by Kruś and Bronisz (2000), we have found a calculation error in a numerical example reported in their article. The characteristic function of this example is shown in the third column of Table 1. The first and second columns of the table show an index $c \in \{1, \ldots, 2^n - 1\}$ that we use to refer to each coalition and the players who conform them, respectively. The next three columns show the correct nucleolus solution x we have computed for this example, and the excess vector in non-decreasing order together with the index of each coalition in this vector. The last three columns show the solution $\bar{x}$ given by Kruś and Bronisz (2000), and the corresponding excess vector.