Mathematical Programming 29 (1984) 323-347
North-Holland

ON THE CORE AND NUCLEOLUS OF MINIMUM COST SPANNING TREE GAMES

Daniel GRANOT*

Faculty of Commerce, University of British Columbia, Vancouver, British Columbia, Canada V6T 1Y8

Gur HUBERMAN

Graduate School of Business, University of Chicago, 5801 S. Ellis Ave., Chicago, IL 60637, USA

Received 12 May 1980
Revised manuscript received 3 August 1983

We develop two efficient procedures for generating cost allocation vectors in the core of a minimum cost spanning tree (m.c.s.t.) game. The first procedure requires $O(n^2)$ elementary operations to obtain each additional point in the core, where n is the number of users. The efficiency of the second procedure, which is a natural strengthening of the first procedure, stems from the special structure of minimum excess coalitions in the core of an m.c.s.t. game. This special structure is later used (i) to ease the computational difficulty in computing the nucleolus of an m.c.s.t. game, and (ii) to provide a geometric characterization for the nucleolus of an m.c.s.t. game. This geometric characterization implies that in an m.c.s.t. game the nucleolus is the unique point in the intersection of the core and the kernel. We further develop an efficient procedure for generating fair cost allocations which, in some instances, coincide with the nucleolus. Finally, we show that by employing Sterns' transfer scheme we can generate a sequence of cost vectors which converges to the nucleolus.

Key words: Cooperative Games, Core, Nucleolus, Kernel, Cost Allocation, Spanning Tree.

1. Introduction

Recently, game theory has been frequently used to model various cost allocation problems, see, for example [2, 3, 7, 8, 12, 13, 15, 16, 21, 23], and [1, 19] for a non-atomic game formulation. In general, a cost allocation problem has the following features. There is a set $N=\{1,\ldots,n\}$ of users (communities, divisions in a corporation, products, etc.) who cooperate in the undertaking of a joint enterprise (service department, cablevision network, emergency system, transportation system, etc.). The question is how to allocate the cost of the joint venture among the set of users so as to satisfy criteria such as fairness, stability, efficiency, minimal dysfunctional inducement, etc.

The game theory formulation represents the cost of undertaking a joint project which caters, separately, only to subsets $S\subset N$ in a characteristic function form.

* Part of this research was done while the author was visiting the Department of Operations Research at Stanford University. This research was partially supported by Natural Sciences and Engineering Research Council Canada Grant A-4181.

Explicitly, for a subset S of N, let $c(S)$ represent the cost of undertaking a similar joint venture which serves only the members of S. A cooperative game in characteristic function form is the pair $(N; c)$. The core of the game consists of all cost allocation vectors x which cover the total cost of the joint project, i.e., $\sum_{j=1}^{n} x_j = c(N)$, and for which no subset of users has an incentive to sever its cooperation with the rest of the users, i.e., $\sum_{j\in S} x_j \leq c(S)$ for each $S \subset N$.

Vectors in the core are natural candidates for 'good' cost allocations. However, a game may have an empty core, and even if the core is not empty, generating vectors in it may be computationally very difficult.

The class of minimum cost spanning tree games is an example of a collection of games which possess a nonempty core, and for which a vector in the core can be generated easily. Minimum cost spanning tree (m.c.s.t.) games arise in cost allocation problems in which the joint enterprise is a tree connecting the agents to a common source. They were studied by Claus and Kleitman [5], Bird [2], and by Granot and Huberman [8]. Megiddo [15] showed that the core of a game in which the graph connecting agents to a common source is a Steiner tree might be empty. Also, Littlechild [12] and Megiddo [16] computed the nucleolus for a special class of m.c.s.t. games, see also [9].

It was shown in [8] that a vector in the core of an m.c.s.t. game, referred to as the L solution, can be simply read from an associated minimum cost spanning tree (m.c.s.t.) graph. However, the L solution has apparent deficiencies. It discriminates against users which are closer to the common supplier while subsidizing the more distant ones, and thus cannot be offered as a fair solution to an m.c.s.t. game. It is desirable therefore to further investigate the structure of the core of an m.c.s.t. game, with the hope of generating efficiently other vectors in it, and to consider other concepts of solution such as the nucleolus. In Section 2 which follows we formally introduce the class of m.c.s.t. games, provide necessary definitions and notation, and explain the deficiencies of the L solution.

In Section 3 we show that the structure of the core of an m.c.s.t. game allows the development of an extremely efficient procedure for generating numerous additional cost allocations in the core. Each vector in the core is generated using $O(n^2)$ elementary operations, where n is the number of users. In Section 4 we develop a second procedure for generating points in the core, which is a natural strengthening of the first procedure. The second procedure is rather efficient. In particular, if the associated m.c.s.t. graph is a chain, it produces each additional point by using $O(n^2)$ elementary operations. The efficiency of the second procedure stems from the special structure of minimum excess coalitions in the core of an m.c.s.t. game (Theorem 4). This special structure is extremely helpful in our investigation of the nucleolus of an m.c.s.t. game. Specifically, in Section 5 we use Theorem 4 to ease the computational burden in calculating the nucleolus. For example, we show that if the associated m.c.s.t. graph is a chain, the number of constraints in the linear programs that are solved in Kopelowitz method [11] (or Maschler, Peleg and Shapley's method [14]) in order to produce the nucleolus of an m.c.s.t. game can be reduced to the order of n^2, compared with 2^n in a general cooperative game.

In Section 6 we employ Theorem 4 to provide a geometric characterization for the nucleolus of an m.c.s.t. game. Specifically, Theorem 8 reveals that in an m.c.s.t. game, the nucleolus is the unique point in the core in which any pair of users which are adjacent in an m.c.s.t. graph is situated symmetrically with respect to its bargaining range. Further, we show (Corollary 3) that in an m.c.s.t. game the nucleolus is the unique point in the intersection of the kernel (of the grand coalition) and the core.

Finally, in Section 7 we develop an efficient procedure for generating fair cost allocations which, in some instances, coincide with the nucleolus. We also show that by employing Sterns' transfer scheme [22], we can construct a sequence of cost vectors which converges to the nucleolus of an m.c.s.t. game.

2. Minimum cost spanning tree games

Our own study of minimum cost spanning tree games was motivated by the discussion of Claus and Kleitman [5] on criteria for allocating costs among users of a spanning tree network (e.g., cities which employ jointly a cablevision network). A definition of minimum cost spanning tree (m.c.s.t.) games and a constructive core nonemptiness result are provided in this section. Since our paper deals with costs (as opposed to revenues), it is natural and convenient to reverse traditional inequalities in cooperative game theory.

A cooperative n-person game is a pair $(N; c)$ where $N=\{1, 2, \ldots, n\}$ and c is a real valued function on the subsets of N with $c(\phi)=0$. The *excess* of a nonempty coalition $S \subset N$ with respect to a vector x is $ex(S, x) \equiv c(S) - x(S)$, where $x(S) = \sum_{i \in S} x_i$. The *core* of $(N; c)$ will be denoted by $C(N; c)$. It is a subset of $\mathbb{R}^n$ defined as $C(N; c) = \{x \in \mathbb{R}^n\colon ex(N, x) = 0$ and $ex(S, x) \geq 0$ for all $S \subset N\}$.

Consider the complete graph K_{n+1} with nodes $\{0, 1, \ldots, n\}$. Node 0 is the common supplier and the set $N=\{1, 2, \ldots, n\}$ is the users' set. The users are identified with players of the m.c.s.t. game. The entries of a symmetric matrix $C = (c_{ij})$ $(i, j = 0, 1, \ldots, n)$ indicate the costs of establishing edges between nodes i and j. For $S \subset N$, the graph $\Gamma_S = (V_S, E_S)$ is an m.c.s.t. whose node set is $V_S = \{0\} \cup S$. The set $E_S \subset \{0\} \cup S \times \{0\} \cup S$ is the edge set of Γ_S. Among all possible edge sets that will connect all nodes in V_S, the set E_S has the smallest total cost, namely $c(S) = \sum_{(i,j) \in E_S} c_{ij}$. The numeral $c(S)$ is thus determined for every $S \subset N$. Hence the construction of the characteristic function of the m.c.s.t. game $(N; c)$.

The tree associated with the grand coalition $\Gamma_N = (V_N, E_N)$ induces a partial order $>$ on $\{0\} \cup N$. Write $i > j$ if node j is on the (unique) path connecting node i and node 0 in Γ_N. Under the order $>$ each node $i \in N$ has one immediate predecessor $p(i)$ and has a set (possibly empty) of immediate followers $F(i)$. Node 0 has a nonempty set of followers, $F(0)$. Denote by V_j the set containing node j and its followers (i.e., $V_j = \{j\} \cup \{i \in N: i > j\}$). Also, denote the edge set $E_j = \{(r, k) \in E_N: r \in V_j \text{ and } k \in V_j\}$. Call the graph $T_j = (V_j, E_j)$ *the subtree of* Γ_N *rooted at node* j. A subset $S \subset V_N$ will be called *Γ_N-connected* if for each $i, j \in S$ there is a subset of edges $E \subset E_N \cap (S \times S)$ which form a path connecting i and j. An equivalent definition is that $S \subset V_N$ is Γ_N-connected if it contains a smallest element and if $i, j \in S$ and $i > j$ imply $\{k \in N: i > k > j\} \subset S$.

The main known result on m.c.s.t. games is the core nonemptiness theorem, namely:

Theorem 1. [8]. *The vector* $L(C) \equiv (l_1, l_2, \ldots, l_n)$ *is in the core of the m.c.s.t. game* $(N; c)$ *where* $l_i = c_{ip(i)}$.

The vector $L(C)$ will be referred to as the minimum spanning tree (m.s.t.) solution or the L solution. Of course, given the cost matrix C one can have more than one m.c.s.t. Γ_N and consequently more than one m.s.t. solution. Observe that the L solutions can be computed without the need to compute the values of the characteristic function $c(\cdot)$. They can be simply read from the various m.c.s.t. graphs associated with the cost matrix (c_{ij}).

Ease of computation is a virtue of the m.s.t. solution. However, something is lacking with the L solution. If the grand coalition forms, each node uses its predecessor to link itself to the common supplier 0. One feels that the value of such a service should be reflected in a cost allocation among the users. This is not the case with the L solution. For instance, the leaves of Γ_N (i.e., nodes with no followers) pay under L the smallest amount they would pay under any cost allocation in the core. This observation generalizes: for each S such that $E_S \subset E_N$, $\sum_{i \in S} l_i = c(S)$. Furthermore, $N \backslash S$ pays under L the least amount it would pay according to any vector in the core. Again, unless $\sum_{i \in N \backslash S} l_i = c(N \backslash S)$, the set $N \backslash S$ is using the set of users S in order to link themselves to the common supplier.

This criticism of the L solution is the motivation for a search for other cost allocations. Preferably, they should be in the core and easy to compute.

3. Weak demand operations

This section presents an extremely efficient procedure which produces various core cost allocations. The procedure is motivated by considering opportunity costs of followers and transferring followers' surplus to their predecessors. The following example illustrates these ideas.

Example 1. Let $N = \{1, 2, \ldots, 6\}$. The m.c.s.t. is illustrated in Figure 1. Arc costs (the c_{ij}'s) are indicated next to the arcs. Missing arcs have infinite costs.

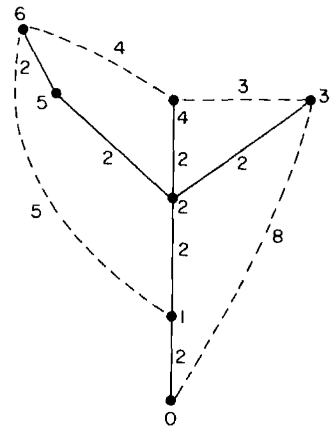

Fig. 1. The data for Example 1. The arcs of the m.c.s.t. Γ_N are the solid lines.

In this example, $\Gamma_N = (\{0, 1, 2, 3, 4, 5, 6\}, \{(0, 1), (1, 2), (2, 3), (2, 4), (2, 5), (5, 6)\})$ and $L = (2, 2, 2, 2, 2, 2)$. If it were not for node 1, the set $\{2, 3, 4, 5, 6\}$ would have to pay 8 (rather than 2) in order to link itself to 0. Suppose user 1 required that the surplus of 6 be transferred to him, at the expense of user 2. Compliance with this demand results in the cost allocation $(-4, 8, 2, 2, 2, 2)$. Suppose that user 2 picks up on the idea and observes that arcs $(2, 3)$, $(2, 4)$ and $(2, 5)$ are used by its followers to connect themselves with 0. The edge set of the m.c.s.t. of $N\backslash\{2\}$ is $\{(0, 1), (1, 6), (6, 5), (6, 4), (3, 4)\}$. If the set N loses the cooperation of user 2, the branch starting with user 3 (containing only 3) will have to pay 3 (as opposed to 2), the branch rooted at 4 will have to pay 4 and the branch rooted at 5 (containing 5 and 6) will pay 5. A successful transfer of the surplus (i.e., $3-2+4-2+5-2=6$) to user 2 (relative to the cost allocation $(-4, 8, 2, 2, 2, 2)$) results in the vector $(-4, 2, 3, 4, 5, 2)$.[1] When user 5 makes a similar demand on user 6 one ends up with the cost allocation $(-4, 2, 3, 4, 3, 4)$. It is straightforward to check that the three cost vectors we generated are in the core of $(N; c)$ of Example 1.

The following formalizes the ideas in Example 1. For a node $i \in N$ consider the m.c.s.t. $\Gamma_{N\backslash\{i\}} = (V_N\backslash\{i\}, E_{N\backslash\{i\}})$ in which the subtrees T_k, $(k \in F(i))$, rooted at k in Γ_N preserve their internal structure,[2] i.e., $E_k \subset E_{N\backslash\{i\}}$. For each $k \in F(i)$ there is one edge $(r, q) \in E_{N\backslash\{i\}}$ such that $r \in V_k$ and q is on the unique path of every node in V_k to 0 in $\Gamma_{N\backslash\{i\}}$. Denote the cost of this edge by c_k. A *weak demand operation by*

[1] Other cost allocations are possible in this case. For example, such an action by user 2 can result with the cost vectors $(-4, 2, 5, 4, 3, 2)$, $(-4, 2, 5, 3, 4, 2)$, $(-4, 2, 3, 5, 4, 2)$ and $(-4, 2, 4, 5, 3, 2)$, all of which are in the core of Example 1. To keep the presentation within limits these cost vectors are not characterized and studied here.

[2] Since the T_k's, $k \in F(i)$, are subtrees of Γ_N, there always exists an m.c.s.t. graph $\Gamma_{N\backslash\{i\}} = (V_{N\backslash\{i\}}, E_{N\backslash\{i\}})$ in which the subtrees T_k, $k \in F(i)$, preserve their internal structure.

user i transforms a cost allocation y into the vector $wd^i(y)$, given by:

$$wd_r^i(y)=\begin{cases} c_k-\left(\sum_{e\in E_k} c(e)-y(V_k\backslash\{k\})\right), & r=k,\ k\in F(i),\\ y_i-\sum_{k\in F(i)}(wd_k^i(y)-y_k), & r=i,\\ y_r, & \text{otherwise},\end{cases} \tag{1}$$

where $c(e)$ is the cost associated with arc e, i.e., if $e=(u,v)$ then $c(e)=c_{uv}$. Note that if $F(i)=\phi$ then $wd^i(y)=y$.

The transformation $y\to wd^i(y)$ is well defined by (1) for each $y\in\mathbb{R}^n$. It satisfies the invariance property $wd^i(wd^j(y))=wd^j(wd^i(y))$ for any i and j, with $wd^i(wd^i(y))=wd^i(y)$. The invariance follows from the observation that the transformation, $y\to wd^i(y)$, involves transfer of costs only between i and members of $F(i)$ and leaves unchanged $y(V_j\backslash\{j\})$, if $j<i$, $j>i$, or j and i are not comparable. Thus, we extend the definition of a weak demand operation to a transformation carried out by a subset Q, $Q\subset N$. A vector, y, is mapped to the vector, $wd^Q(y)$, where $wd^{\emptyset}(y)=y$. If $Q\neq\emptyset$, take any $i\in Q$ to define recursively $wd^Q(y)=wd^i(wd^{Q\backslash\{i\}}(y))$.

We first prove:

Theorem 2. *Let $\Gamma_N=(V_N,E_N)$ be an m.c.s.t. graph with an associated m.s.t. solution L. For every $i\in N$ the vector $wd^i(L)$ is in the core of the associated m.c.s.t. game.*

Proof. Since $L\in C[(N;c)]$ and $\sum_{r\in S} wd_r^i(L)\leq\sum_{j\in S} l_j$ if $i\in S$, it suffices to show that $\sum_{r\in S} wd_r^i(L)\leq c(S)$ if $i\notin S$. Assume, on the contrary, that for some S, with $i\notin S$, we have that $\sum_{r\in S} wd_r^i(L)>c(S)$. Then $\sum_{r\in N\backslash\{i\}} wd_r(L)>c(S)+\sum_{r\in N\backslash(S\cup\{i\})} wd_r^i(L)$. By definition of L and $wd^i(L)$ we have that $\sum_{r\in N\backslash\{i\}} wd_r^i(L)$ is the cost associated with an m.c.s.t. graph $\Gamma_{N\backslash\{i\}}=(V_{N\backslash\{i\}},E_{N\backslash\{i\}})$. Moreover, $c(S)+\sum_{r\in N\backslash(S\cup\{i\})} wd_r^i(L)$ is the cost associated with a spanning tree on $V_{N\backslash\{i\}}$. Therefore, $\sum_{r\in N\backslash\{i\}} wd_r^i(L)>c(S)+\sum_{r\in N\backslash(S\cup\{i\})} wd_r^i(L)$ contradicts the minimality of $\Gamma_{N\backslash\{i\}}$, and the proof of Theorem 2 follows.

Moreover,

Theorem 3. *For any $Q\subset N$, $wd^Q(L)\in C[(N;c)]$.*

Proof. See Appendix.

Observe that since an m.c.s.t. graph over n nodes is constructed in $O(n^2)$ elementary operations, a new cost allocation vector $wd^i(wd^S(L))$ is produced from $wd^S(L)$, $S\subseteq N$, by $O(n^2)$ elementary operations.

We remark that Theorem 3 cannot be generalized to hold for an arbitrary vector y in the core and an arbitrary m.c.s.t. graph Γ_N. Indeed, consider the following example.

Example 2. Consider the m.c.s.t. graph shown in Figure 2. The L solution of the associated m.c.s.t. game is $L=(1,1,1,1,1)$. It is easy to show that $y=(-1,-3,6,2,1)$ is in the core of the associated m.c.s.t. game, and that $wd^4((-1,-3,6,2,1))=(-1,-3,6,-1,4)$. But, $(-1,-3,6,-1,4)$ is not in the core since $c(\{3,5\})=9<wd_3^4(y)+wd_5^4(y)=6+4=10$.

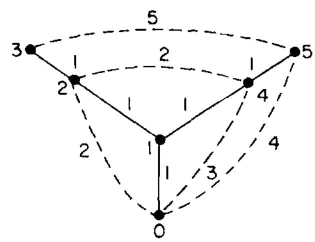

Fig. 2. The data for Example 2.

However, in the next section we will show that if $\Gamma_N=(V_N, E_N)$ is a chain and $y\in C[(N;c)]$ then $wd^S(y)$ is in the core for any $S\subseteq N$.

Finally, it is assumed in the definition of $wd^i(y)$ that user i performs a simultaneous w.d.o. against the entire set, $F(i)$, of his followers. However, one can similarly define a partial w.d.o. of user i against a subset $\hat{F}(i)$, $\hat{F}(i)\subseteq F(i)$, of his followers. In such a w.d.o. user i makes the cost associated with e_{ij}, $j\in\hat{F}(i)$ excessively high, and thus forces the users in $\hat{F}(i)$ to pay the cost incurred to them if they get the service from the central supplier not directly through user i. Such an action by user i will produce other vectors in the core. For example, in Example 1, starting from the vector $(-4,8,2,2,2,3)$ and letting user 2 perform partial weak demand operations against various subsets of his followers one can get the vectors $(-4,7,3,2,2,3)$, $(-4,7,2,3,2,3)$, $(-4,6,2,2,4,3)$, $(-4,5,3,4,2,3)$, $(-4,5,3,2,4,3)$ and $(-4,2,3,4,5,3)$, all of which are contained in the core of the m.c.s.t. game associated with Example 1. For more details see [10].

4. Strong demand operations

We develop in this section a second procedure for generating points in the core, which is a natural strengthening of the first procedure presented in Section 3. The computational efficiency of the second procedure stems from the special structure of minimum excess coalitions in the core of an m.c.s.t. game (Theorem 4). This special structure is used later (Sections 5 and 6) to investigate the properties of the nucleolus of an m.c.s.t. game.

The following example will motivate the subsequent developments.

Example 3. Let $N=\{1,2,3\}$ and let the corresponding m.c.s.t. graph $\Gamma_{\{1,2,3\}}$ be given in Figure 3 below.

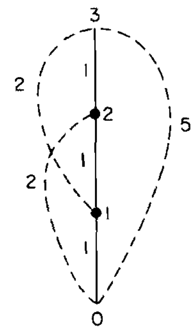

Fig. 3. The data for Example 3.

The L vector of the associated m.c.s.t. game is $L=(1,1,1)$. Using notation of the previous section we have $wd^1((1,1,1))=(0,2,1)$, and $wd^2((0,2,1))=(0,1,2)$. However, it is conceivable that user 2 may require from user 3 to pay more than 2, the reason being that, while indeed it costs user 3 to join user 1 only 2, user 1 is also being subsidized by user 2. In fact, in the cost allocation vector $(0,2,1)$ user 1 does not pay anything, and it is user 2 who is paying the cost of delivering a service from the common supplier for both users 1 and 2. Thus, user 2 may require user 3 to pay such an amount for which, if user 3 joins user 1, both users 1 and 3 pay the cost associated with the edge set of the m.c.s.t. graph $\Gamma_{\{1,3\}}$. This implies that user 3 must pay 3, which results with the cost vector $(0,0,3)$. It is easy to check that $(0,0,3)$ is in the core of the m.c.s.t. game associated with Example 3. Moreover, the vector $(0,0,3)$ cannot be generated by any sequence of weak demand operations.

The formalization of strong demand operations entails the following notation. For subsets R_1 and R_2 of N and a cost vector y let

$$T_{R_1,R_2}=\{S\colon R_1\subseteq S, R_2\cap S=\emptyset\},$$

$$\mathcal{S}_{R_1,R_2}(y)=\{S\colon S\in T_{R_1,R_2}, ex(S,y)\leq ex(R,y), R\in T_{R_1,R_2}\}.$$

(For convenience, if both R_1 and R_2 are singleton, say $R_1=\{i\}$ and $R_2=\{j\}$, we will use the notation $T_{i,j}$ and $\mathcal{S}_{i,j}(y)$ for $T_{\{i\},\{j\}}$ and $\mathcal{S}_{\{i\},\{j\}}(y)$.)

When a node $i\in N$ performs a strong demand operation (s.d.o.) it receives a transfer z_j from each j in its immediate follower set $F(i)$. Explicitly, given a cost allocation vector y in the core, an s.d.o. by user i is a multifunction (correspondence) which associates with y the set of vectors $\{sd^i(y)\}$ given by

$$\{sd^i(y)\}=\left\{x\colon \begin{bmatrix} x_k=y_k+z_k, & k\in F(i), \\ x_k=y_i-\sum_{j\in F(i)} zj, & k=i, \\ x_k=y_k, & \text{otherwise}, \end{bmatrix}\right\} \tag{2}$$

where $z_j = x_j - y_j$, $j \in F(i)$, and the x_j's are the optimal values of the t_j variables, $j \in F(i)$, in the following optimization problem:

$$\text{Max} \quad \{\sum_{j \in F(i)} t_j\} \tag{3}$$

$$\begin{aligned} \text{s.t.} \quad & ex(R, t) \geq 0 \quad \text{for all } R \in T_{F(i),\{i\}} \cup (T_{S,F(i)\setminus S}\colon S \subset F(i)), \\ & t_k = y_k \quad \text{for } k \not\in \{i\} \cup F(i), \\ & t(N) = y(N). \end{aligned} \tag{4}$$

If y is in the core then the constraints in (4) ensure that all cost allocation vectors x, $x \in \{sd^i(y)\}$, are also contained in the core. Further, it is evident from (3)–(4) that $\sum_{j \in F(i)} z_j = \sum_{j \in F(i)} (x_j - y_j)$, $x \in \{sd^i(y)\}$, is the maximum total transfer that user i can get from $F(i)$ without violating core constraints.

Observe that if we impose in (4) the additional restrictions $t_j = y_j$, $j \in \hat{F}(i) \subset F(i)$, then $\{sd^i(y)\}$ can be considered as the collection of points in the core which may result from an s.d.o. performed by user i against only the subset $F(i)\setminus\hat{F}(i)$ of his followers. Further, if $F(i) = \emptyset$ then $\{sd^i(y)\} = \{y\}$.

The main shortcoming of the formulation (3)–(4) of the s.d.o. is its computational impracticality, the linear program has many constraints. Considerable simplification could be achieved by identifying constraints in (4) which will never be binding. It turns out that Γ_N-connected sets play a major role in the elimination of redundant constraints in (4). This will be stated formally as Proposition 1 in the sequel. In fact, it follows from Proposition 1 that if Γ_N is a chain then $\{sd^i(y)\}$ (which is a singleton set in this case) can be computed in $O(n^2)$ elementary operations (see Corollary 1 below).

Indeed, let us first consider the case where Γ_N is a chain, and assume that the nodes of Γ_N are numbered $0, 1, \ldots, n$ so that $1 \leq i < j \leq n$ implies that i is closer than j to node 0. It follows from Proposition 1 (or Corollary 1) that an s.d.o. performed by user i transforms a cost vector y into a vector $sd^i(y)$, given by

$$sd^i_j(y) = \begin{cases} \text{Min}_{r=0,\ldots,i-1} \left\{ ex(\{1, \ldots, r\}, y) + c_{uv} - \left(\sum_{e \in E_{i-1}} c(e) - y(V_{i+2}) \right) : \right. \\ \qquad\qquad \left. u \leq r,\ v \geq i+1 \right\}, & j = i+1, \\ y_i - (sd^i_{i+1}(y) - y_{i+1}), & j = 1, \\ y_j, & \text{otherwise} \end{cases} \tag{5}$$

where $ex(\emptyset, y) = 0$.

It clearly follows from (5) that in order to compute $sd^i(y)$ we will consider at most i arcs (u, v), $u \leq i-1$, $v \geq i+1$. Therefore, $sd^i(y)$ can be computed in $O(n^2)$ elementary operations.

To proceed with the general case we need the following definition. Given a set $Q \subset N$, the pair $P(Q) = (V(Q), E(Q))$ is an *open connected subgraph of Γ_N induced* by Q if (i) $V(Q) \subset N$, $E(Q) \subset E_N$, (ii) $(V(Q) \cup Q, E(Q))$ is the smallest connected subgraph of Γ_N whose node set contains Q, and (iii) $E_N \backslash E(Q) \subset (V_N \backslash V(Q)) \times (V_N \backslash V(Q))$ (i.e. the complement of $P(Q)$ with respect to Γ_N is a legitimate graph.)

Note that a set $Q \subset N$ can induce more than one open connected subgraph. Moreover, $V(Q) \cap Q$ need not be empty. The following is an example of open connected subgraphs.

Example 4. Consider the m.c.s.t. graph shown in Figure 4. The sets $(\{2,3\}, \{(1,2),(2,3),(2,4)\})$ and $(\{2\}, \{(1,2),(2,3),(2,4)\})$ are open connected subgraphs induced by the nodes 1, 3 and 4.

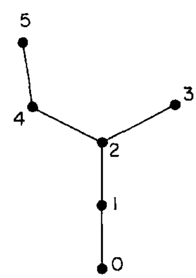

Fig. 4. The m.c.s.t. for Example 4.

With each cost allocation y and an open connected subgraph $P(Q) = (V(Q), E(Q))$, we associate the *marginal excess*

$$mex(P(Q), y) \equiv \sum_{(i,j) \in E(Q)} c_{ij} - y(V(Q)). \tag{6}$$

Also, for the subtree $T_j = (V_j, E_j)$ of Γ_N rooted at j we define its marginal excess (with respect to the cost vector y) as $mex(T(j), y) = \sum_{(k,r) \in E_j} d_{kr} - y(V_j \backslash \{j\})$. The following lemmas provide estimates on these marginal excesses, given that y is in the core.

Lemma 1. *If $y \in C(N; c)$ then*

$$mex(T(i), y) \leqslant 0. \tag{7}$$

Proof. Since $y \in C(N; c)$ we have

$$y(N \backslash \{V_i \backslash \{i\}\}) \leqslant c(N \backslash \{V_i \backslash \{i\}\}) = c(N) - \sum_{(k,r) \in E_i} c_{kr}. \tag{8}$$

Also (since $y(N) = c(N)$) we have

$$c(N) - y(N \backslash \{V_i \backslash \{i\}\}) = y(V_i \backslash \{i\}). \tag{9}$$

The addition of (8) and (9) completes the proof.

Lemma 2. *Let $P(Q) = (V(Q), E(Q))$, $0 \notin V(Q)$, be an open connected subgraph of Γ_N induced by Q, and let E be a collection of arcs such that $E \cap E_N = \emptyset$ and*

$(V_N \backslash V(Q), \{E_N \backslash E(Q)\} \cup E)$ *is a connected graph. Then*

$$mex(P(Q), y) \leqslant \sum_{(k,r) \in E} c_{kr} \tag{10}$$

for every $y \in C[(N; c)]$.

Proof. By the connectedness of $(V_N \backslash V(Q), \{E_N \backslash E(Q)\} \cup E)$ and since $0 \notin V(Q)$ we have that

$$c(N \backslash V(Q)) \leqslant \sum_{(r,k) \in E_N \backslash E(Q)} c_{rk} + \sum_{(r,k) \in E} c_{rk}.$$

Since y is in the core we have $y(N \backslash V(Q)) \leqslant c(N \backslash V(Q))$, which implies

$$y(N \backslash V(Q)) \leqslant \sum_{(r,k) \in E_N \backslash E(Q)} c_{rk} + \sum_{(r,k) \in E} c_{rk}. \tag{11}$$

Also, $y \in C(N; c)$ implies

$$y(N) = \sum_{(r,k) \in E_N} c_{rk}. \tag{12}$$

Subtract (11) from (12) to obtain

$$y(V(Q)) \geqslant \sum_{(r,k) \in E(Q)} c_{rk} - \sum_{(r,k) \in E} c_{rk},$$

thus completing the proof.

Our next target is the establishment of an effective necessary condition for a coalition to have minimal excess. To this end, consider a set $S \subset N$. Its complement $N \backslash S$ can be uniquely partitioned into maximal subsets of $N \backslash S$, each of which is Γ_N-connected. This collection of sets will be denoted by $\mathscr{P}_S$. For $V \in \mathscr{P}_S$, let $E^V = \{(i, j) \in E_N : i \in V \text{ or } j \in V\}$. The Γ_N-connectedness of V guarantees that (V, E^V) is an open connected subgraph of Γ_N.

The following theorem shows that for core members, the minimum excess coalitions in an m.c.s.t. game are few and easily identified. This theorem will be used in the sequel to show how the computation of cost allocations resulted from strong demand operations can be simplified, and to investigate the properties of the nucleolus of an m.c.s.t. game.

Theorem 4. (Minimum excess coalition structure Theorem). *Let* $\Gamma_N = (V_N, E_N)$ *be an m.c.s.t. graph and let* $S, R \subset N$ *be such that* $\mathscr{P}_S \subset \mathscr{P}_R$. *Then* $ex(S, y) \leqslant ex(R, y)$ *for any* $y \in C[(N; c)]$.

Proof. Let $R, S \subset N$ be such that $\mathscr{P}_S \subset \mathscr{P}_R$. Recall that $\Gamma_R = (V_R, E_R)$ is an m.c.s.t. with $V_R = R \cup \{0\}$. Our proof of Theorem 4 uses Procedure I below.

Procedure I.

Step 0. Set $W = R$, $E = E_R$ and $\mathscr{P} = \mathscr{P}_R \backslash \mathscr{P}_S$.

Step 1. If $\mathscr{P} = \emptyset$, terminate. Otherwise go to Step 2.

Step 2. Let $Q \in \mathscr{P}$. If $Q \neq V_j \backslash \{j\}$ for some subtree T_j of Γ_N rooted at j, go to Step 3. Otherwise, set $W = W \cup Q$, $E = E \cup E^Q$ and $\mathscr{P} = \mathscr{P} \backslash \{Q\}$. (Note that, by Lemma 1, $ex(W \cup Q, y) \leq ex(W, y)$.) Go to Step 1.

Step 3. Let $F \subset E \backslash E_N$ be the (unique) collection of arcs such that the graphs $(V_N \backslash Q, \{E_N \backslash E^Q\} \cup F)$ and $(\{0\} \cup W \cup Q, \{E \cup E^Q\} \backslash F)$ are connected. Set $W = W \cup Q$, $E = E \cup E^Q$ and $\mathscr{P} = \mathscr{P} \backslash Q$. (Note that by Lemma 2 $mex((Q, E^Q), y) \leq \sum_{(i,j) \in F} c_{ij}$, which implies that $ex(W \cup Q, y) \leq ex(W, y)$.) Go to Step 1.

Procedure I terminates after $|\mathscr{P}_R \backslash \mathscr{P}_S|$ iterations, with $ex(W, y) \leq ex(R, y)$ and $W = S$, thus completing the proof.

Theorem 4 leads naturally and implies immediately:

Proposition 1. *Let* $y \in C(N; c)$ *and* $S \subset F_{(i)}$.

(i) *There exists a coalition* $R \in \mathscr{S}_{F(i),\{i\}}(y)$ *such that* $N \backslash R$ *is* Γ_N*-connected.*

(ii) *If* $i \not\in Q$ *for some* $Q \in \mathscr{S}_{S,F(i) \backslash S}(y)$, *then there exists a coalition* $R \in \mathscr{S}_{S,\{F(i) \backslash S\}}(y)$ *such that* $N \backslash R$ *is* Γ_N*-connected.*

(iii) *If* $i \in Q$ *for some* $Q \in \mathscr{S}_{S,F(i) \backslash S}(y)$, *then there exists a coalition* $R \in \mathscr{S}_{S,F(i) \backslash S}(y)$ *such that* $\mathscr{P}_R$ *has* $|F(i) \backslash S|$ *members, and each node* j, $j \in F(i) \backslash S$, *is the smallest element in precisely one member of* $\mathscr{P}_R$.

The importance of Proposition 1 is that it limits the search for members of $\mathscr{S}_{S,F(i) \backslash S}(y)$ (and of $\mathscr{S}_{F(i),\{i\}}(y)$) to a small group of candidates. Moreover, these candidates have a well defined graphical structure which is specified in Proposition 1. Finally, due to their strcuture, not only their identification but also the computation of $c(R)$ for $R \in \mathscr{S}_{S,F(i) \backslash S}(y)$ (or $R \in \mathscr{S}_{F(i),\{i\}}(y)$) can be efficiently done.

Example 5. Consider the m.c.s.t. shown in Figure 5 below, and assume that a cost allocation vector y is in the core of the associated m.c.s.t. game. Suppose

$$R = \{2, 4, 6, 7, 9, 12, 14, 15, 17, 18, 20\} \in \mathscr{S}_{20,19}(y),$$

and that

$$\Gamma_R = (R \cup \{0\}, \{d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8, d_9, d_{10}, d_{11}\}).$$

By Lemma 1, $mex(P(20), y) \leq 0$. Further, by Lemma 2 we have:

(i) $mex(P(\{0, 2\}), y) \leq c(d_1)$,

(ii) $mex(P(\{4, 6, 7\}), y) \leq c(d_3) + c(d_4)$,

(iii) $mex(P(\{12, 14, 15\}), y) \leq c(d_8) + c(d_9)$,

(iv) $mex(P(15, 17), y) \leq c(d_{11})$, and

(v) $mex(P(\{2, 4, 9, 12, 18\}), y) \leq c(d_2) + c(d_5) + c(d_6) + c(d_7)$. All the above inequalities imply that $S = \{1, 2, \ldots, 24\} \backslash \{19\} \in \mathscr{S}_{20,19}(y)$. Observe that $N \backslash S = \{19\}$ is Γ_N-connected.

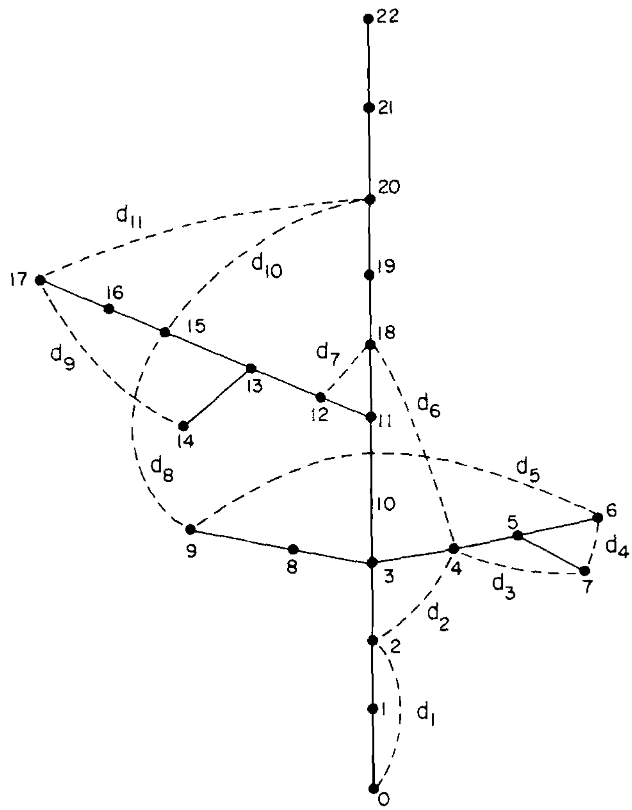

Fig. 5. The m.c.s.t. for Example 5.

Further, consider Example 6 below.

Example 6. Consider the m.c.s.t. graph shown in Figure 6.

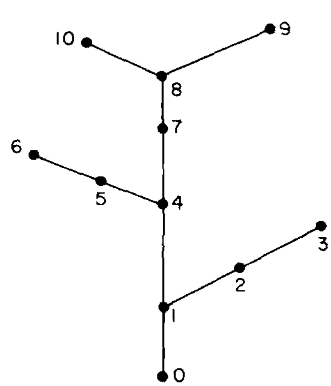

Fig. 6. The m.c.s.t. for Example 6.

Now, let y be a cost allocation vector in the core of the associated m.c.s.t. game, and assume that we seek a subset $S \in \mathscr{S}_{\{9,10\},\{8\}}(y)$, i.e., we seek a subset S, $S \in T_{\{9,10\},\{8\}}$ for which $ex(S, y) \leq ex(R, y)$, for all $R \in T_{\{9,10\},\{8\}}$. In view of Proposition 1, the search for such a subset S can be confined to those subsets Q, $Q \in T_{\{9,10\},\{8\}}$, whose complements $N \backslash Q$ are Γ_N-connected. In Example 6 there are only 13 such subsets, which are: $\{9,10\}$, $\{6,9,10\}$, $\{3,9,10\}$, $\{3,6,9,10\}$, $\{2,3,9,10\}$, $\{5,6,9,10\}$, $\{3,5,6,9,10\}$, $\{2,3,5,6,9,10\}$, $\{1,2,3,9,10\}$, $\{1,2,3,6,9,10\}$, $\{1,2,3,5,6,9,10\}$, $\{1,2,3,4,5,6,9,10\}$, and $\{1,2,3,4,5,6,7,9,10\}$. On the other hand, the total number of subsets in $T_{\{9,10\},\{8\}}$ in Example 6 is $2^7 = 128$.

Let us consider now the special case where Γ_N is a chain. It follows immediately from Proposition 1:

Corollary 1. *If $\Gamma_N = (V_N, E_N)$ is a chain then for every cost allocation y in $C[(N; c)]$ and for every i, $i \in N$, $sd^i(y)$ given by (5) is in the core of $(N; c)$.*

Assume still that Γ_N is a chain, and let $R = \{i_1, \ldots, i_k\}$ be an ordered subset of elements of N, such that $i_j \not\in V_t$ for $j = 1, \ldots, k-1$ and $t > j$, $t \in R$. Strong demand operations carried out by the users in R, performed according to the order in R, transforms the cost vector y to a vector $sd^R(y)$, defined recursively as $sd^R(y) = sd^{i_k}(sd^{R \backslash \{i_k\}}(y))$, where $sd^{\emptyset}(y) = y$. It follows immediately from Corollary 1 that if y is in the core then $sd^R(y)$ is in the core for any subset R. However, by contrast with the weak demand operation, $sd^R(y)$ is not invariant to the order in which s.d.o. are carried out by members of R. Further, if y is in the core then $sd^{\{1,\ldots,n-1\}}(y) = sd^{\{1,\ldots,n-1\}}(L)$, where L is the m.s.t. solution associated with Γ_N.

Finally, recalling Example 2, we know that if Γ_N is not a chain then, even if y is in the core, $wd^i(y)$ is not necessarily in the core. However, by the definitions of $sd^i(y)$ and $wd^i(y)$ and from Corollary 1 it follows:

Corollary 2. *If $\Gamma_N = (V_N, E_N)$ is a chain and if y is in the core then $wd^S(y)$ is in the core for any subset $S \subseteq \{1, \ldots, n-1\}$.*

5. Computing the nucleolus of an m.c.s.t. game

The calculation of the nucleolus of a cooperative game is usually a computationally difficult task. An algorithm for this purpose was designed by Kopelowitz [11] and Maschler et al. [14]. In this section we apply the minimum excess coalition structure theorem (i.e., Theorem 4) to reduce significantly the computational effort involved in generating the nucleolus of an m.c.s.t. game.

Before presenting the section's main result, let us recall the following definitions. For a cost allocation $x \in \mathbb{R}^n$, let $\theta(x) \in \mathbb{R}^{2^n}$ be the vector whose entries are the excesses $ex(S, x)$, $S \subset N$, arranged in an increasing order. Denoting by $\geqslant$ the lexicographic order in $\mathbb{R}^{2^n}$, define the nucleolus of a game $Nu(N; c)$ as

$$Nu(N; c) = \{x \in \mathbb{R}^n\colon \theta(x) \geqslant \theta(z) \text{ for all } z \text{ such that } x(N) = z(N) = c(N)\}.$$

Schmeidler [20] showed that the nucleolus of $(N; c)$ is a unique point in the kernel of $(N; c)$. Moreover, if the core of $(N; c)$ is not empty, the nucleolus is in it.

Kopelowitz [11] and Maschler et al. [14] have constructed an algorithm for computing the nucleolus of a cooperative game by solving a sequence of at most 2^n linear programs, where n is the number of players. The first linear program P_1 to be solved in this algorithm is Problem P_1:

$$Max\{r, \text{s.t. } c(N) = x(N), r \leqslant c(S) - x(S), S \in \xi_0\},$$

where $\xi_0 = 2^N \backslash \{N, \emptyset\}$.

For $j \geqslant 1$, let r_j denote the optimal value of r in problem P_j;

$$A_j = \{x\colon x \text{ is an optimal solution to problem } P_j\};$$

$$\xi_j = \left\{S\colon S \in \xi_0 \quad \left\{\bigcup_{k=1}^{j-1} \xi_k\right\}, r_j = c(S) - x(S) \text{ for all } x \in A_j\right\}.$$

At stage i, the linear programming problem to be solved is problem P_i:

$$\begin{aligned} &Max \quad r \\ &\text{s.t.} \quad r_j = c(S) - x(S), \quad S \in \xi_j, j = 1, 2, \ldots, i-1, \\ &\qquad r \leqslant c(S) - x(S), \quad S \in \xi_0 \quad \left\{\bigcup_{j=1}^{i-1} \xi_j\right\}, \\ &\qquad x(N) = c(N). \end{aligned}$$

The nucleolus is obtained in stage t, $1 \leqslant t < 2^n$, whenever A_t consists of a single vector.

The main result in this section is Theorem 5 which follows.

Theorem 5. *Let $(N; c)$ be an m.c.s.t. game with an associated m.c.s.t. $\Gamma_N = (V_N, E_N)$. The nucleolus of $(N; c)$ depends only on the coalitions S whose complements $N \backslash S$ are Γ_N-connected.*

Proof. We will show that in the sequence of linear programming problems P_i that are solved to produce the nucleolus, the constraints corresponding to subsets S whose complements $N\backslash S$ are not Γ_N-connected are redundant. Let z denote the nucleolus of the m.c.s.t. game $(N; c)$. Assume that for some k, $k \geqslant 1$, and for some $S \subset N$,

$$r_k = c(S) - z(S), \quad S \in \xi_k, \tag{13}$$

and $N\backslash S$ is not Γ_N-connected. Let m be the smallest integer such that $R_1, \ldots, R_m$ is a collection of pairwise disjoint maximal Γ_N-connected sets such that $\bigcup_{j=1}^m R_j = N\backslash S$. Observe that $m \geqslant 2$. Since the nucleolus is always contained in the core, if the core is not empty, it follows from Theorem 4 that

$$ex(N\backslash R_i, z) \leq ex(S, z), \quad i = 1, \ldots, m. \tag{14}$$

However, (14) implies that the constraint corresponding to the subset $N\backslash R_i$ is tight at problem $P_{t(i)}$, i.e.,

$$r_{t(i)} = c(N\backslash R_i) - z(N\backslash R_i), \quad i = 1, \ldots, m \tag{15}$$

and, most important, $t(i) \leq k$. Therefore, by the construction of the linear programs P_i it follows that the system (15) is a subset of the constraint set in problem P_k.

Now, since $z(N\backslash R_i) = c(N) - z(R_i)$, the system (15) implies that

$$z(N\backslash S) = \sum_{i=1}^{m} z(R_i) = m \cdot c(N) + \sum_{i=1}^{m} [r_{t(i)} - c(N\backslash R_i)] \tag{16}$$

or, equivalently,

$$z(S) = \sum_{i=1}^{m} [c(N\backslash R_i) - r_{t(i)}] - (m-1)c(N). \tag{17}$$

Clearly, we must have from (13) and (17) that

$$c(S) - r_k = \sum_{i=1}^{m} [c(N\backslash R_i) - r_{t(i)}] - (m-1)c(N). \tag{18}$$

Thus the constraint (13), which is satisfied at problem P_k, is implied by the system of constraints (15) which are also satisfied at problem P_k. However, the system of constraints (15) correspond to subsets S whose complements $N\backslash S$ are Γ_N-connected. Therefore, (13) is redundant and the proof of Theorem 5 is complete.

It follows from Theorem 5 that the number of constraints in the linear programs P_i that we need to solve to produce the nucleolus of an m.c.s.t. game is much smaller than that in the problems P_i we need to solve to compute the nucleolus of a general cooperative game. In particular, if Γ_N is a chain, the number of constraints in the first linear programming problem that has to be solved to produce the nucleolus is only $\frac{1}{2}n(n+1)$, compared with 2^n in a general cooperative game.

6. A geometric characterization of the nucleolus of an m.c.s.t. game

This section provides a geometric characterization of the nucleolus of an m.c.s.t. game. This geometric characterization implies that for m.c.s.t. games the intersection of the core and the kernel consists of a unique point, which is the nucleolus. First, we show:

Theorem 6. *Let $(N; c)$ be an m.c.s.t. game with an associated m.c.s.t. $\Gamma_N = (V_N, E_N)$.*

Then, a cost allocation vector y in the core of $(N; c)$ *is the nucleolus of* $(N; c)$ *if and only if*

$$Min\{ex(S, y), S \in T_{i,j}\} = Min\{ex(S, y), S \in T_{j,i}\} \tag{19}$$

for all nodes i, j *such that* $(i, j) \in E_N$.

Proof. Clearly, if (19) is not satisfied then y is not contained in the kernel, see also [6, 14]. However, the nucleolus is always contained in the kernel and thus y is not the nucleolus. It remains to show that if (19) holds then y is the nucleolus.

Let $\xi_0(y) = \{\emptyset, N\}$ and for $i \geq 1$ let

$$\xi_i(y) = \left\{ S\colon ex(S, y) \leq ex(Q, y),\ S, Q \in 2^N \Big\backslash \bigcup_{j=0}^{i-1} \xi_j(y) \right\}.$$

Let Q be an arbitrary subset in $\xi_1(y)$. By Theorems 4 and 5 we may assume that $N \backslash Q$ is Γ_N-connected. We will first use Procedure II below to show that any decrease in $y(Q)$, with an equal increase in $y(N \backslash Q)$ will produce a lexicographically inferior vector than y. Clearly, an increase in $y(Q)$ will certainly produce a lexicographically inferior vector to y.

We define the following sets.

$$U = \{(k, l)\colon k \notin Q, l \in Q \cap F(k)\}, \qquad V = \{k\colon k \notin Q \text{ and } p(k) \in Q\}.$$

(Recall that $p(k)$ is the immediate predecessor of k in Γ_N.)

Procedure II.

Step 1. If $U = \emptyset$ go to Step 2. Otherwise, let $(i, j) \in U$, and let $R \in \mathcal{S}_{i,j}(y)$. We have:

(i) $R \in \xi_1(y)$, since $Q \in \xi_1(y)$ and (19) imply that $ex(Q, y) = ex(R, y)$,

(ii) $N \backslash Q \subset R$, by Theorem 4,

(iii) $j \in N \backslash R \subset V_j \subset Q$, where the first inclusion follows from the definition of R, and the last two inclusions follow from Theorem 4.

Observe that (i), (ii) and (iii) above imply that any decrease in $y(N \backslash R)$, and an equal increase in $y(N \backslash Q)$ will produce a lexicographically inferior vector to y.

Now, if $N \backslash R = V_j$ let $U = U \backslash \{i, j\}$ and go to Step 1. If, on the other hand, $N \backslash R$ is a proper subset of V_j, let

$$U_R = \{(k, l)\colon k \in N \backslash R \text{ and } l \in R \cap F(k)\},$$

set $U = U \cup U_R \backslash \{(i, j)\}$, and go to Step 1.

Step 2. If $V = \emptyset$ stop. Otherwise, let $i \in V$ and let $R \in \mathcal{S}_{i,p(i)}(y)$. We have:

(i) $R \in \xi_1(y)$, since $Q \in \xi_1(y)$ and (19) imply that $ex(R, y) = ex(Q, y)$,

(ii) $N \backslash Q \subset R$, by Theorem 4. In fact, Theorem 4 implies that $N \backslash Q \subset V_i \subset R$,

(iii) $p(i) \in N \backslash R \subset N \backslash V_i$, where the first inclusion follows from the definition of R and the second inclusion follows from Theorem 4.

Again, observe that (i), (ii) and (iii) above imply that any decrease in $y(N \backslash R)$, and an equal increase in $y(N \backslash Q)$, will produce a lexicographically inferior vector to y.

Now, if $N \backslash R = N \backslash V_i$ let $V = V \backslash \{i\}$ and go to Step 2. If, on the other hand, $N \backslash R$ is a proper subset of $N \backslash V_i$, let

$$V_R = \{k: k \in N \backslash R \text{ and } p(k) \in R\}, \qquad U_R = \{(k, l): k \in N \backslash R \text{ and } l \in R \cap F(k)\}.$$

(Observe that $V_R \cup U_R \subseteq \{N \backslash V_i\} \backslash \{p(i)\}$.) Set $U = U_R$, and $V = V \cup V_R \backslash \{i\}$, and go to Step 1.

Procedure II will terminate after a finite number of iterations with the conclusion that any decrease in $y(Q)$, with an equal increase in $y(N \backslash Q)$, will produce a lexicographically inferior vector to y. Similarly, we can show that for an arbitrary subset W, $W \in \xi_2(y)$, any decrease in $y(W)$, with an equal increase in $y(N \backslash W)$, will produce a lexicographically inferior vector to y, and so forth. Eventually, the characteristic vectors of all subsets Q whose excesses we cannot change, by a local change in y, without producing a lexicographically inferior vector will span $\mathbb{R}^n$. This implies that any local change in the cost vector y, to produce another cost vector x, will result with x being lexicographically inferior to y. However, the nucleolus is an optimal solution to a linear programming problem, see e.g. [4, 17], in which every local optimum is also a global optimum. It follows, therefore, that y is the nucleolus of the associated m.c.s.t. game, and the proof of Theorem 6 follows.

Let $\mathcal{K}[(N; c)]$ denote the kernel, of the grand coalition, of an m.c.s.t. game. An immediate result that follows from Theorem 6 is:

Corollary 3. *In an m.c.s.t. game the nucleolus is the unique point in* $C[(N; c)] \cap \mathcal{K}[(N; c)]$.

Proof. If y is in the intersection of the kernel and the core then (19) must hold, see also [14]. However, by Theorem 6, y must be the nucleolus, which is unique, and the proof of Corollary 3 follows.

An interesting geometric characterization for the intersection of the kernel and the core of a cooperative game was given by Maschler, Peleg and Shapley [14]. Let us briefly review their geometric characterization, as applied to an m.c.s.t. game, and relate it to Theorem 6 and Corollary 3. First we need some notation. Let x be in $C[(N; c)]$. For any pair of users i, j, $i \neq j$, let $s_{i,j}(x) = ex(S, x)$, $S \in \mathcal{S}_{i,j}(x)$, and let

$$\delta_{i,j} = Max\{\delta: x + \delta u^i - \delta u^j \in C[(N; c)]\},$$

where u^i represents the ith unit vector. Observe that $\delta_{i,j}(x) = s_{i,j}(x)$. Further, let $R_{i,j}(x)$ be the line segment with end points

$$(x - s_{i,j}(x) u^i + s_{j,i}(x) u^j,\ x + s_{i,j}(x) u^i - s_{j,i}(x) u^j).$$

As observed in [14], $R_{i,j}(x)$ can be regarded as the bargaining range between users i and j with respect to x, and the middle point of $R_{i,j}(x)$ represents a situation in which both players are symmetric with respect to their bargaining range. Now, Theorem 3.8(a) in [14], applied to m.c.s.t. games, yields:

Theorem 7. *Let $(N; c)$ be an m.c.s.t. game. If $x \in C[(N; c)]$ then x belongs to $\mathcal{K}[(N; c)] \cap C[(N; c)]$ if, and only if, for each $i, j \in N$, $i \neq j$, x bisects the line segment $R_{i,j}(x)$.*

However, Theorem 6 and Corollary 3 imply that in an m.c.s.t. game the geometric characterization of $\mathcal{K}[(N; c)] \cap C[(N; c)]$ is much simpler, namely:

Theorem 8 (A geometric characterization of the nucleolus of an m.c.s.t. game). *Let $(N; c)$ be an m.c.s.t. game with an associated m.c.s.t. graph $\Gamma_N = (V_N, E_N)$. If $x \in C[(N; c)]$ then x belongs to $\mathcal{K}[(N; c)] \cap C[(N; c)]$ if, and only if, for each $(i, j) \in E_N$, x bisects the line segment $R_{i,j}(x)$. Moreover, the nucleolus is the unique point x in the core which bisects $R_{i,j}(x)$ for each $(i, j) \in E_N$.*

It follows from Theorem 8 that the nucleolus of an m.c.s.t. game $(N; c)$, with an associated m.c.s.t. graph Γ_N, is the unique cost allocation vector in the core in which every pair (i, j) of adjacent users in Γ_N (i.e. $(i, j) \in E_N$) is situated symmetrically with respect to its bargaining range.

7. Fair cost allocations and the nucleolus

In Section 5 we used the minimum excess coalition structure theorem to reduce the computational burden involved in computing the nucleolus. However, one still has to solve a sequence of linear programs (albeit smaller, due to Theorem 5) in order to produce the nucleolus. For large n, this may be computationally prohibitive.

Motivated by the geometric characterization of the nucleolus (Theorem 8), and by the strong demand operation, we develop in this section an efficient procedure for generating fair cost allocation vectors in the core which, in some instances, coincide with the nucleolus. When Γ_N is a chain this procedure generates a unique point, denoted by $f(L)$, using $O(n^3)$ elementary operations. Further, we are able to provide easy necessary and sufficient conditions for $f(L)$ to be the nucleolus of the associated m.c.s.t. game. In the event that a cost vector generated by our procedure does not coincide with the nucleolus, it can be improved upon by using Sterns' transfer scheme [22]. In fact, we show that by using Sterns' results we can generate a sequence of cost allocation vectors which converges to the nucleolus of an m.c.s.t. game.

Our procedure for generating fair cost vectors is as follows. The users start from the L solution[3] and act in a nondecreasing order. A user i, at his turn, requires from his followers $F(i)$ to adjust the current cost allocations of $\{i\} \cup F(i)$ so as to produce a new cost allocation vector y for which

$$s_{i,j}(y) = s_{j,i}(y), \quad j \in F(i). \tag{20}$$

[3] One can similarly define a procedure in which the users start from any cost vector in the core. However, Theorem 9 which follows is valid only when we start from the L solution.

Recall that in accordance with the strong demand operation, each user requires from his followers to collectively pay the maximum, without violating core restraints. However, Theorem 6 (or 8) suggests that the above approach for readjusting a cost allocation vector, as reflected by (20), is more appropriate if we seek fair cost allocations.

To further clarify the above idea consider again the m.c.s.t. game associated with Example 3, whose corresponding m.c.s.t. graph Γ_N is reproduced in Figure 7.

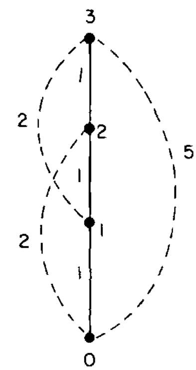

Fig. 7. The data of Example 3.

The associated L solution is $L=(1,1,1)$. Readjusting only the allocations to users 1 and 2 results with the cost allocation vector $x=(\frac{1}{2}, 1\frac{1}{2}, 1)$, for which $s_{1,2}(x)= s_{2,1}(x)=\frac{1}{2}$. Similarly, readjusting only the allocations to users 2 and 3 produces the vector $y=(\frac{1}{2}, \frac{3}{4}, 1\frac{3}{4})$, for which $s_{2,3}(y)=s_{3,2}(y)=\frac{3}{4}$. It is easy to verify that $y=(\frac{1}{2}, \frac{3}{4}, 1\frac{3}{4})$ is, in fact, the nucleolus of the associated m.c.s.t. game.

In general, let $\Gamma_N=(V_N, E_N)$ be an m.c.s.t. graph with an associated m.s.t. solution L. Assume first that Γ_N is a chain, and that the nodes in V_N are ordered so that $1 \leqslant i<j \leqslant n$ if i is closer than j to 0 in Γ_N. For $i=1,\dots,n$ let $M(i)=\{1,\dots,i\}$, and define the vector $f^{M(1)}(L)$ as follows:

$$f_j^{M(1)}(L)=\begin{cases} l_1-\frac{1}{2}(sd_2^1(L)-l_2), & j=1,\\ l_2+\frac{1}{2}(sd_2^1(L)-l_2), & j=2,\\ l_j, & j\geqslant 2,\end{cases}$$

where $sd^1(L)$ is the vector derived from L by an s.d.o. carried out by user 1.

The vector $f^{M(i+1)}(L)$ is defined recursively as follows:

$$f_j^{M(i+1)}(L)=\begin{cases} f_j^{M(i)}(L), & j\leqslant i,\\ f_{i+1}^{M(i)}(L)-\frac{1}{2}[sd_{i+2}^{i+1}(f^{M(i)}(L))-l_{i+2}], & j=i+1,\\ l_{i+2}+\frac{1}{2}[sd_{i+2}^{i+1}(f^{M(i)}(L))-l_{i+2}], & j=i+2,\\ l_j, & j\geqslant i+3.\end{cases}$$

For simplicity of notation we let $f(L)=f^{M(n-1)}(L)$. Observe that in Example 3 $f(L)=(\frac{1}{2},\frac{3}{4},1\frac{3}{4})$.

Clearly, since $sd^i(y)$ can be constructed from y, $y\in C[(N;c)]$, in $O(n^2)$ elementary operations, $f(L)$ can be constructed using $O(n^3)$ elementary operations. Moreover, by Corollary 1 and since $L\in C[(N;c)]$, $f(L)\in C[(N;c)]$. We will provide below easy necessary and sufficient conditions for $f(L)$ to be the nucleolus.

Theorem 9. *$f(L)$ is the nucleolus of the associated m.c.s.t. game if and only if*

$$f_{i+1}^{M(i)}(L)\leq c_{pq},\qquad q\geq i+1,\quad p\leq i-1,\quad i=1,\ldots,n-1. \tag{21}$$

Proof. First observe that by the construction of $f(L)$ we have that

$$ex(M(i),f(L))=s_{i+1,i}(f(L)),\quad i=1,\ldots,n-1. \tag{22}$$

Therefore, by Theorem 6, $f(L)$ is the nucleolus if, and only if,

$$ex(M(i),f(L))=s_{i,i+1}(f(L)),\quad i=1,\ldots,n-1. \tag{23}$$

From Theorem 4 we have that for each i there exists a subset $Q^i\in\mathscr{S}_{i,i+1}(f(L))$ of the form $Q^i=M(i)\cup A_r$, where A_r is either empty or of the form $A_r=\{r,r+1,\ldots,n\}$, $r\geq i+2$.

Now, if (20) is satisfied for each i, then

$$ex(M(i),f(L))\leq ex(Q^i,f(L)),\quad i=1,\ldots,n-1, \tag{24}$$

and (23) follows. If on the other hand, for some $\hat{i}$ and some $\hat{p},\hat{q}$, $\hat{q}\geq\hat{i}+1$, $\hat{p}\leq\hat{i}-1$, $f_{i+1}^{M(\hat{i})}(L)>c_{\hat{p}\hat{q}}$, then

$$\begin{aligned}s_{\hat{p},\hat{p}+1}(f(L))&\leq ex(M(\hat{p})\cup\{i+1,\ldots,n\},f(L))\\&<ex(M(\hat{p}),f(L))=s_{\hat{p}+1,\hat{p}}(f(L)).\end{aligned} \tag{25}$$

It follows from Theorem 6 that in this case $f(L)$ is not the nucleolus, and the proof of Theorem 9 follows.

We will consider now the case where $\Gamma_N=(V_N,E_N)$ is not necessarily a chain. A permutation σ of N will be referred to as a *nondecreasing order in Γ_N* provided that $\sigma(i)<\sigma(j)$ if j is not on the unique path from i to 0 in Γ_N. We will denote by $f_\sigma(L)$ the cost allocation vector derived from the L solution after all the users, following a nondecreasing order σ, are acting as follows. If, e.g., $\sigma(i)=k$ and $F(i)\neq\emptyset$, then at stage k user i demands from $F(i)$ to adjust the current allocation of $\{i\}\cup F(i)$ to produce another cost allocation vector y for which $s_{i,j}(y)=s_{j,i}(y)$, $j\in F(i)$. Clearly, all nodes j, $j\in N$, for which $F(j)=\emptyset$ are dummy users in the above procedure for modifying the cost vector.

Different nondecreasing orders may produce different cost allocation vectors, as can be seen from the following example.

Example 7. Consider the m.c.s.t. graph depicted in Figure 8.

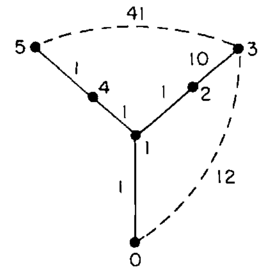

Fig. 8. The m.c.s.t. for Example 7.

The L solution of the associated m.c.s.t. game is $L=(1,1,10,1,1)$. Further, there are essentially two nondecreasing orders, $\sigma_1=\{1,2,4,3,5\}$ and $\sigma_2=\{1,4,2,3,5\}$ and it is easy to show that

$$f_{\sigma_1}((1,1,10,1,1))=(-41,5\tfrac{1}{2},11,16,22\tfrac{1}{2})$$

and

$$f_{\sigma_2}((1,1,10,1,1))=(-41,5\tfrac{1}{2},11,16\tfrac{1}{2},22).$$

Using Theorem 6, one can easily verify that

$$f_{\sigma_1}((1,1,10,1,1))=(-41,5\tfrac{1}{2},11,16,22\tfrac{1}{2})$$

is, in fact, the nucleolus of the m.c.s.t. game associated with Example 7. However, the fact that

$$f_{\sigma_2}((1,1,10,1,1))=(-41,5\tfrac{1}{2},11,16\tfrac{1}{2},22)$$

is not the nucleolus reveals that conditions like (21), generalized in a natural way to the case where Γ_N is not a chain, can only provide necessary conditions for $f_\sigma(L)$ to be the nucleolus. Of course, Theorem 6 (or 8) always provide us with both necessary and sufficient conditions for any $y\in C[(N;c)]$ to be the nucleolus of $(N;c)$.

If $f_\sigma(L)$ is not the nucleolus, for some nondecreasing order σ, it can be improved upon by employing Sterns' transfer scheme [22]. To show that, let us introduce the following notation. For $i,j\in N$, $i\neq j$, and $x\in C[(N;c)]$ let $k_{i,j}(x)=\frac{1}{2}(s_{i,j}(x)-s_{j,i}(x))$,

$$I(x)=\{(i,j)\colon k_{i,j}(x)\leq k_{u,v}(x),\ u,\ v\in N,\ u\neq v\},$$

and $k(x)=k_{i,j}(x)$ for some $(i,j)\in I(x)$. It follows from Theorem 6 that if x is the nucleolus then $k(x)=0$. Now, following Sterns [22], we say that y results from x by a *transfer of maximal size* if, and only if, $y_i=x_i+k(x)$, $y_j=x_j-k(x)$, and $y_r=x_r$, $r\not\in\{i,j\}$ for some $(i,j)\in I(x)$.

From [22, Lemma 2] we have that $k(x)\leq k(y)$. Moreover, we can prove,

Theorem 10. *Let $\{x^i\}$ be a sequence of vectors for $i\geq 0$ such that $x^0\in C[(N;c)]$ and each x^{i+1} results from x^i by a transfer of maximal size. Then $\{x^i\}$ converges to the nucleolus of $(N;c)$.*

Proof. Since $x^0 \in C[(N;c)]$, and by the definition of a transfer of maximal size, $x^i \in C[(N;c)]$ for $i \geq 0$. By Theorem 3 in [22], $\{x^i\}$ converges to a point in the kernel. However, by Theorem 6, the unique point in $\mathcal{K}[(N;c)] \cap C[(N;c)]$ is the nucleolus, and the proof follows.

For general cooperative games, Sterns reports that the convergence to points in the kernel has been satisfactory. Therefore, in view of the availability of good starting points $f_\sigma(L)$, the above transfer scheme can be used to efficiently generate the nucleolus of an m.c.s.t. game.

Acknowledgement

The authors acknowledge some very useful remarks made by two anonymous referees on an earlier draft of this paper.

Appendix

Proof of Theorem 3. We use induction on the cardinality $|Q|$ of the subset Q. Theorem 2 asserts the result for $|Q| = 1$, and we assume that result is valid for any m.c.s.t. game $(N;c)$ in which $|Q| \leq q-1$. Now, let q be the cardinality of Q. Let $i \in Q$ be maximal within Q with respect to the order $>$. Since $wd^Q(y)$ is independent of the order in which w.d.o. are performed by members of Q we have that $wd^Q(L) = wd^i(wd^{Q\setminus\{i\}}(L))$. Let $z = wd^{Q\setminus\{i\}}(L)$. By the induction hypothesis $z \in C[(N;c)]$. Let $F(i) = \{j_1, \ldots, j_t\}$ and let $T_{j_1} = (V_{j_1}, E_{j_1}), \ldots, T_{j_t} = (V_{j_t}, E_{j_t})$ be the subtrees of Γ_N rooted at $j_1, \ldots, j_t$, respectively. Let $\Gamma_{N\setminus\{i\}}$ be an m.c.s.t. in which $T_{j_1}, \ldots, T_{j_t}$ maintain their internal structure (see footnote 2). Let e_k be the unique edge in $\Gamma_{N\setminus\{i\}}$ whose one extremity is in V_{j_k}, the other extremity in $V_{N\setminus\{i\}} \setminus V_{j_k}$ and which is on the unique path of every user in V_{j_k} to 0 in $\Gamma_{N\setminus\{i\}}$. Further, let c_k denote the cost associated with e_k. We will assume, for simplicity of exposition, that arc e_k joins T_{k-1} and T_k, $(k = 1, \ldots, t)$ in $\Gamma_{N\setminus\{i\}}$, where T_0 is the subtree of Γ_N induced by

$$V_{N\setminus\{i\}} \Big\backslash \left\{ \bigcup_{k=1}^{t} V_{j_k} \right\}.$$

By definition of the vector $wd^i(z)$, we have that $\sum_{r \in S} wd^i_r(z) \leq \sum_{r \in S} z_r$ if $i \in S$. Thus, in order to show that $wd^i(z) \in C[(N;c)]$ it suffices to show that $\sum_{r \in S} wd^i_r(z) \leq c(S)$ when $i \notin S$. Next, we consider the m.c.s.t. game determined by the symmetric cost matrix $\tilde{C} = (\tilde{c}_{ij})$ given below,

$$\tilde{c}_{rv} = \begin{cases} c_1 & \text{if } r = i \text{ and } v = j_1 \text{ or } r = j_1 \text{ and } v = 1, \\ M & \text{if } r = i \text{ and } v = j_k,\ k = 2, \ldots, t, \text{ or } r = j_k,\ k = 2, \ldots, t, \\ & \quad \text{and } v = i, \\ c_k & \text{if } r = j_k \text{ and } v = j_{k-1},\ k = 2, \ldots, t, \text{ or } r = j_{k-1} \\ & \quad \text{and } v = j_k,\ k = 2, \ldots, t, \\ c_{rv} & \text{otherwise,} \end{cases}$$

where M is a large enough number so that (i, j_k), $k = 2, \ldots, t$, is not in any m.c.s.t. graph associated with $(\tilde{c}_{rv})$. The m.c.s.t. graph $\tilde{\Gamma}_N$ associated with the cost matrix $(\tilde{c}_{ij})$ is identical to the original m.c.s.t. graph Γ_N except for the following changes: arc (j_k, i) in Γ_N is replaced by arc (j_{k-1}, j_k) in $\tilde{\Gamma}_N$ for $k = 2, \ldots, t$. The L solution to the m.c.s.t. game $(N; \tilde{c})$ is $L(\tilde{C}) = (\tilde{l}_1, \ldots, \tilde{l}_n)$, where $\tilde{l}_r = c_r$ for $r = j_1, \ldots, j_t$ and $\tilde{l}_r = l_r$ otherwise. Now, when the members in $Q \backslash \{i\}$ employ w.d.o. in the m.c.s.t. graph $\tilde{\Gamma}_N$ we obtain the vector $\tilde{z}$ which, by the induction hypothesis, satisfies $\tilde{z} \in C[(N; \tilde{c})]$. Since there is no j in $Q \backslash \{i\}$ for which $j > i$, the w.d.o. performed by $Q \backslash \{i\}$ did not affect any rth component of the $\tilde{L}$ vector, $r > i$. Moreover, by the definition of the w.d.o. (given in (1)), w.d.o. performed by $Q \backslash \{i\}$ in $(N; c)$ are identical to those performed by $Q \backslash \{i\}$ in $(N; \tilde{c})$. We therefore have

$$\tilde{z}_r = \begin{cases} c_r, & \text{for } r = j_k \ (k = 1, \ldots, t), \\ z_r, & \text{otherwise.} \end{cases}$$

By our induction assumption $\tilde{z}(S) \leq \tilde{c}(S)$ for all $S \subset N$. Further, when $i \notin S$ $\tilde{c}(S) \leq c(S)$ since $\tilde{c}_{uv} \leq c_{uv}$ for all u and v such that $u \neq i$, $v \neq i$. Moreover, by the construction of $\tilde{C}$ and the definition of $wd^i(z)$ we have that $wd^i_r(z) = \tilde{z}_r$ for $r \neq i$. Therefore, for any S such that $i \notin S$ we have $\sum_{r \in S} wd^i_r(z) = \sum_{r \in S} \tilde{z}_r \leq \tilde{c}(S) \leq c(S)$, and the proof follows.

References

[1] L.J. Billera, D.C. Heath and J. Raanan, "Internal telephone billing rates—A novel application of non atomic game theory", *Operations Research* 26 (1978) 956–965.

[2] C.G. Bird, "On cost allocation for a spanning tree: A game theory approach", *Networks* 6 (1976) 335–350.

[3] J. Callen, "Financial cost allocations. A game theoretic approach", *Accounting Review* 53 (1978) 303–308.

[4] A. Charnes and K. Kortanek, "On classes of convex and preemptive nuclei for N-person games", in: H.W. Kuhn, ed., *Proceedings of the 1967 Princeton Mathematical Programming Symposium*, Princeton University Press (Princeton, 1970).

[5] A. Claus and D.J. Kleitman, "Cost allocation for a spanning tree", *Networks* 3 (1973) 289–304.

[6] M. Davis and M. Maschler, "The kernel of a cooperative game", *Naval Research Logistics Quarterly* 12 (1965) 223–259.

[7] D. Granot, "On the role of cost allocation in locational models", Faculty of Commerce and Business Administration, University of British Columbia (January 1981).

[8] D. Granot and G. Huberman, "On minimum cost spanning tree games", *Mathematical Programming* 21 (1981) 1–18.

[9] D. Granot and G. Huberman, "The relationship between convex games and minimum cost spanning tree games: A case for permutationally convex games", *SIAM Journal of Algebraic and Discrete Methods* 3 (1982) 288–292.

[10] D. Granot and G. Huberman, "More cost allocations in the core of a minimum cost spanning tree game", Working paper No. 695, Faculty of Commerce, University of British Columbia (February 1980, revised December 1981).

[11] A. Kopelowitz, "Computation of the kernels of simple games and the nucleolus of N-person games", Research Memorandum No. 31, Department of Mathematics, The Hebrew University (Jerusalem, September 1967).

[12] S.C. Littlechild, "A simple expression for the nucleolus in a special case", *International Journal of Game Theory* 3 (1974) 21–29.

[13] S.C. Littlechild and G. Owen, "A simple expression for the Shapley value in a special case", *Management Science* 20 (1973) 370–372.

[14] M. Maschler, B. Peleg and L.S. Shapley, "Geometric properties of the kernel, nucleolus, and related solution concepts", *Mathematics of Operations Research* 4 (4) (1979) 303–338.

[15] N. Megiddo, "Cost allocation for Steiner trees", *Networks* 1 (1979) 1–9.

[16] N. Megiddo, "Computational complexity and the game theory approach to cost allocation for a tree", *Mathematics of Operations Research* 3 (1978) 189–196.

[17] G. Owen, "A note on the nucleolus", *International Journal of Game Theory* 3 (1974) 101–103.

[18] A. Roth and R. Verrecchia, "The Shapley value as applied to cost allocation: A re-interpretation", *Journal of Accounting Research* 17 (1979) 295–303.

[19] D. Samet, Y. Tauman and I. Zang, "An application of the A–S prices for cost allocation in transportation problems", *Mathematics of Operations Research*, forthcoming.

[20] D. Schmeidler, "The nucleolus of a characteristic function game", *SIAM Journal of Applied Mathematics* 17 (1969) 1163–1170.

[21] M. Shubik, "Incentives, decentralized control, the assignment of joint costs and internal pricing", *Management Science* 8 (1962) 325–343.

[22] R.E. Sterns, "Convergent transfer schemes for N-person games", Technical Information Series, No. 67-C-311, R&D Center, General Electric (September 1967).

[23] A. Tamir, "On the core of cost allocation games defined on location problems", Department of Statistics, Tel-Aviv University (July 1980).