

Sharing Supermodular Costs

Andreas S. Schulz

Sloan School of Management and Operations Research Center, Massachusetts Institute of Technology,
Cambridge, Massachusetts 02139, schulz@mit.edu

Nelson A. Uhan

School of Industrial Engineering, Purdue University, West Lafayette, Indiana 47907, nuhan@purdue.edu

We study cooperative games with supermodular costs. We show that supermodular costs arise in a variety of situations; in particular, we show that the problem of minimizing a linear function over a supermodular polyhedron—a problem that often arises in combinatorial optimization—has supermodular optimal costs. In addition, we examine the computational complexity of the least core and least core value of supermodular cost cooperative games. We show that the problem of computing the least core value of these games is strongly NP-hard and, in fact, is inapproximable within a factor strictly less than 17/16 unless P = NP. For a particular class of supermodular cost cooperative games that arises from a scheduling problem, we show that the Shapley value—which, in this case, is computable in polynomial time—is in the least core, while computing the least core value is NP-hard.

Subject classifications: games/group decisions: cooperative; analysis of algorithms: computational complexity; mathematics: combinatorics.

Area of review: Special Issue on Computational Economics.

History: Received September 2007; revisions received April 2008, February 2009, November 2009; accepted March 2010. Published online in *Articles in Advance* July 1, 2010.

1. Introduction

Cooperative game theory offers a mathematical framework for determining “agreeable” ways of sharing the costs collectively incurred by a group of cooperating agents. A (*transferable utility*) *cooperative game* (N, v) is defined by a finite set N of players, and a cost function $v: 2^N \rightarrow \mathbb{R}$, with $v(\emptyset) = 0$. A subset $S \subseteq N$ is referred to as a *coalition*. The quantity $v(S)$ is the joint cost incurred by the agents in coalition S if they cooperate. Cooperative game theory has been used extensively to study cost sharing for a myriad of application areas of operations research (e.g., Owen 1975, Bird 1976, Granot and Huberman 1981, Kalai and Zemel 1982, Potters et al. 1991, Hartman et al. 2000, Goemans and Skutella 2004, Chen and Zhang 2006).

In this work, we focus on situations in which agents face *supermodular*, or increasing marginal, costs. A set function $v: 2^N \rightarrow \mathbb{R}$ is supermodular if

$$v(S \cup \{j\}) - v(S) \leq v(S \cup \{j, k\}) - v(S \cup \{k\}) \quad (1)$$

for all $j, k \in N$ such that $j \neq k$, and $S \subseteq N \setminus \{j, k\}$. We focus on cooperative games (N, v) where v is nonnegative and supermodular. We call such games *supermodular cost cooperative games*.

One of the most important solution concepts in cooperative game theory is the *core* (Gillies 1959). Suppose $x \in \mathbb{R}^N$ is a cost allocation vector: for each $i \in N$, x_i is the cost allocated to agent i . (For notational convenience, for any vector $x \in \mathbb{R}^N$ we define $x(S) = \sum_{i \in S} x_i$ for any $S \subseteq N$.) The core

of a cooperative game (N, v) is the set $\{x \in \mathbb{R}^N: x(N) = v(N), x(S) \leq v(S) \text{ for all } S \subseteq N\}$. In other words, the core of (N, v) is the set of all cost allocations that distribute $v(N)$ —the cost incurred when all agents cooperate—in a way such that no subset of agents would be better off by abandoning the rest of the agents and acting on its own. An empty core can be seen as an indication that cooperation amongst all agents is undesirable.

It is straightforward to show that the core of a supermodular cost cooperative game is empty (as long as costs are not modular).¹ Intuitively, this makes sense: the marginal cost associated with adding a particular agent increases as the size of a coalition grows, diminishing the appeal of cooperation. Even though in this situation cooperation may be undesirable from the perspectives of the individual agents, an external party (e.g., a governing authority) may still be interested in encouraging or enforcing cooperation if the agents’ failure to cooperate causes negative externalities. In this case, one might ask, “How much do we need to penalize a coalition for acting independently in order to encourage all the agents to cooperate?” This notion is captured in the *least core value* of a cooperative game. The *least core* of a cooperative game (N, v) is the set of cost allocations $x \in \mathbb{R}^N$ that are optimal solutions to the linear program

$$z^* = \min \{z: x(N) = v(N), x(S) \leq v(S) + z \text{ for all } S \subseteq N, S \neq \emptyset, N\} \quad (\text{LC})$$

(Shapley and Shubik 1966, Maschler et al. 1979). The optimal value z^* of (LC) is the least core value of (N, v) . The computational complexity of computing a cost allocation in the least core has been studied previously in several contexts (Faigle et al. 2000, Kern and Paulusma 2003, Faigle et al. 2001). Properties of the least core value, on the other hand, seem to have been largely ignored; one exception is Deng (1998).

In this work, we demonstrate that supermodular costs arise in a variety of situations: in particular, we show that the problem of minimizing a linear function over a supermodular polyhedron has supermodular optimal costs (§2). This situation arises often in combinatorial optimization, especially in scheduling. In addition, we study the computational complexity of the least core and least core value of supermodular cost cooperative games. We show that the problem of computing the least core value of these games is strongly NP-hard, and that no ρ -approximation algorithm² for this problem with $\rho < 17/16$ is possible, unless P = NP (§3). We also examine a particular class of supermodular cost cooperative games that arises from a scheduling problem and show that the Shapley value³—which, in this case, is computable in polynomial time—is in the least core of these games, although computing the least core value of these games is NP-hard (§4).

2. A Class of Optimization Problems with Supermodular Optimal Costs

We begin by providing some motivation for looking at cooperative games with supermodular costs. The problem of minimizing a linear function over a *supermodular polyhedron*—a polyhedron of the form $\{x \in \mathbb{R}^N: x(S) \geq u(S) \text{ for all } S \subseteq N\}$, where $u: 2^N \rightarrow \mathbb{R}$ is supermodular—arises in many areas of combinatorial optimization, especially in scheduling. For example, Wolsey (1985) and Queyranne (1993) showed that the convex hull of feasible completion time vectors on a single machine is a supermodular polyhedron. Queyranne and Schulz (1995) showed that the convex hull of feasible completion time vectors for unit jobs on parallel machines with nonstationary speeds is a supermodular polyhedron. The scheduling problem they considered includes various classical scheduling problems as special cases. Goemans et al. (2002) showed that for a scheduling environment consisting of a single machine and jobs with release dates, the convex hull of mean busy time vectors of preemptive schedules is a supermodular polyhedron.

In this section, we show that the optimal value of minimizing a linear function over a supermodular polyhedron is a supermodular function. As a result, by studying supermodular cost cooperative games, we are able to gain insight into the sharing of optimal costs for a wide range of situations.

THEOREM 1. *Let N be a finite set, and let $u: 2^N \rightarrow \mathbb{R}$ be a supermodular function. If $d_j \geq 0$ for all $j \in N$, then the function $v: 2^N \rightarrow \mathbb{R}$ defined by*

$$v(S) = \min \left\{ \sum_{j \in S} d_j x_j: x(A) \geq u(A) \text{ for all } A \subseteq S \right\} \quad \text{for all } S \subseteq N \quad (2)$$

is supermodular.

PROOF. Let S be a subset of N with s elements, and let $j, k \in S$ such that $j \neq k$. Without loss of generality, we assume that $S = \{1, \dots, j-1, j, j+1, \dots, k-1, k, k+1, \dots, s\}$, and that the associated costs satisfy $d_1 \geq \dots \geq d_s$. Define $d_0 = d_{s+1} = 0$, $S^i = \{1, \dots, i\}$ for $i = 1, \dots, s$, and $S^0 = \emptyset$.

It is well known that minimizing a linear function over a supermodular polyhedron can be achieved by a greedy procedure (Edmonds 1970). In particular, the value of $v(S)$ is

$$\begin{aligned} v(S) &= \sum_{i=1}^s d_i (u(S^i) - u(S^{i-1})) \\ &= \sum_{i=1}^s d_i u(S^i) - \sum_{i=0}^{s-1} d_{i+1} u(S^i) \\ &= \sum_{i=0}^s (d_i - d_{i+1}) u(S^i). \end{aligned}$$

We also have that

$$\begin{aligned} v(S \setminus \{l\}) &= \sum_{i=0}^{l-1} (d_i - d_{i+1}) u(S^i) + \sum_{i=l}^s (d_i - d_{i+1}) u(S^i \setminus \{l\}) \\ &\quad \text{for } l = j, k, \\ v(S \setminus \{j, k\}) &= \sum_{i=0}^{j-1} (d_i - d_{i+1}) u(S^i) + \sum_{i=j}^{k-1} (d_i - d_{i+1}) u(S^i \setminus \{j\}) \\ &\quad + \sum_{i=k}^s (d_i - d_{i+1}) u(S^i \setminus \{j, k\}). \end{aligned}$$

Therefore, the effect of adding k to $S \setminus \{k\}$ is

$$\begin{aligned} v(S) - v(S \setminus \{k\}) &= \sum_{i=0}^s (d_i - d_{i+1}) u(S^i) - \sum_{i=0}^{k-1} (d_i - d_{i+1}) u(S^i) \\ &\quad - \sum_{i=k}^s (d_i - d_{i+1}) u(S^i \setminus \{k\}) \\ &= \sum_{i=k}^s (d_i - d_{i+1}) (u(S^i) - u(S^i \setminus \{k\})). \end{aligned}$$

Similarly, the effect of adding k to $S \setminus \{j, k\}$ is

$$\begin{aligned} v(S \setminus \{j\}) - v(S \setminus \{j, k\}) &= \sum_{i=k}^s (d_i - d_{i+1}) (u(S^i \setminus \{j\}) - u(S^i \setminus \{j, k\})). \end{aligned}$$

Since u is supermodular, we have that $u(A) - u(A \setminus \{k\}) \geq u(A \setminus \{j\}) - u(A \setminus \{j, k\})$ for any $A \subseteq N$, $j, k \in A$. This, with the fact that $d_i - d_{i+1} \geq 0$ for all $i = 1, \dots, s$, implies that

$$\begin{aligned} v(S) - v(S \setminus \{k\}) &= \sum_{i=k}^s (d_i - d_{i+1})(u(S^i) - u(S^i \setminus \{k\})) \\ &\geq \sum_{i=k}^s (d_i - d_{i+1})(u(S^i \setminus \{j\}) - u(S^i \setminus \{j, k\})) \\ &= v(S \setminus \{j\}) - v(S \setminus \{j, k\}). \end{aligned}$$

Therefore, v is supermodular. \square

Using similar techniques, we can also show that maximizing a nonnegative linear function over a *submodular polyhedron*—a polyhedron of the form $\{x \in \mathbb{R}^N: x(S) \leq u(S) \text{ for all } S \subseteq N\}$ where $u: 2^N \rightarrow \mathbb{R}$ is submodular⁴—has submodular optimal values. An important example of maximizing a nonnegative linear function over a submodular polyhedron is finding a maximum weight independent set of a matroid; in fact, a version of our result has been mentioned in the literature for this special case (see Nemhauser and Wolsey 1988, p. 715).

3. Computational Complexity

We now turn our attention to the computational complexity of computing the least core value of an arbitrary supermodular cost cooperative game (N, v) . For the remainder of the paper, we assume that there are at least two agents ($n \geq 2$).

THEOREM 2. *Computing the least core value of supermodular cost cooperative games is strongly NP-hard, even if a cost allocation in the least core is known.*

PROOF. We show that any instance of the strongly NP-hard maximum cut problem on an undirected graph (Garey et al. 1976) can be reduced to an instance of computing the least core value of a supermodular cost cooperative game. Consider an arbitrary undirected graph $G = (N, E)$. Let $\kappa: 2^N \rightarrow \mathbb{R}$ be the *cut function* of G ; that is, $\kappa(S) = |\{\{i, j\} \in E: i \in S, j \in N \setminus S\}|$. Also, let the function $\eta: 2^N \rightarrow \mathbb{R}$ be defined as $\eta(S) = |\{\{i, j\} \in E: i \in S, j \in S\}|$. Clearly, η is nonnegative. Using the increasing marginal cost characterization of supermodularity (1), it is straightforward to see that η is supermodular. Using counting arguments, it is also straightforward to show that $\eta(S) + \eta(N \setminus S) + \kappa(S) = \eta(N)$ for any $S \subseteq N$.

Now consider the supermodular cost cooperative game (N, v) , where $v(S) = 2\eta(S)$ for all $S \subseteq N$. For each player $i \in N$, we define the cost allocation $x_i = \deg(i)$, where $\deg(i)$ denotes the degree of node i in G . In addition, let $z = \max_{S \subseteq N, S \neq \emptyset, N} \kappa(S)$. Note that $x(N) = \sum_{i \in N} \deg(i) = v(N)$, and for all $S \subseteq N$, $S \neq \emptyset, N$,

$$z \geq \kappa(S) = (2\eta(S) + \kappa(S)) - 2\eta(S) = x(S) - v(S).$$

Therefore, (x, z) is a feasible solution to (LC). Now suppose (x^*, z^*) is an optimal solution to (LC). Adding the

inequalities $x^*(S) \leq v(S) + z^*$ and $x^*(N \setminus S) \leq v(N \setminus S) + z^*$ for any $S \subseteq N$, $S \neq \emptyset, N$, and using the equality $x^*(N) = v(N)$, we have that

$$2z^* \geq v(N) - v(S) - v(N \setminus S) = 2\kappa(S)$$

for all $S \subseteq N$, $S \neq \emptyset, N$.

Therefore, $z^* \geq z$. It follows that $z^* = z = \max_{S \subseteq N, S \neq \emptyset, N} \kappa(S)$, and x is a cost allocation in the least core of (N, v) . In other words, finding the least core value of (N, v) is equivalent to finding the value of a maximum cut in $G = (N, E)$. \square

In our proof of the above theorem, we show that for any instance of the maximum cut problem on an undirected graph, there exists a supermodular cost cooperative game whose least core value is exactly equal to the value of the maximum cut. Since the maximum cut problem is inapproximable within a factor of $17/16 - \varepsilon$ for any $\varepsilon > 0$ unless $P = NP$ (Håstad 2001), we immediately obtain the following inapproximability result.

COROLLARY 1. *There is no ρ -approximation algorithm for computing the least core value of supermodular cost cooperative games, where $\rho < 17/16$, unless $P = NP$.*

4. A Special Case from Single-Machine Scheduling

In this section, we study a particular supermodular cost cooperative game that arises from scheduling situations. Consider a setting where each agent has a job that needs to be processed on a machine (or processor), and any coalition of agents can potentially open their own machine. Suppose each agent $i \in N$ has a job with processing time $p_i \in \mathbb{R}_{>0}$ and weight $w_i \in \mathbb{R}_{\geq 0}$. Jobs are independent, and are scheduled non-preemptively on a single machine, which can process at most one job at a time. A *scheduling game* is a cooperative game (N, v) where the cost $v(S)$ to a coalition S is the minimum sum of weighted completion times of jobs in S . If weight w_i is interpreted as agent i 's per-unit-time waiting cost, then $v(S)$ can be seen as the minimum total waiting cost for agents in S . By Theorem 1 and the previously mentioned result of Wolsey (1985) and Queyranne (1993), scheduling games are indeed supermodular cost cooperative games. The least core value of scheduling games has a natural interpretation: it is the minimum amount we need to charge any coalition for opening a new machine in order to encourage cooperation.

Cooperative games that arise from scheduling situations have been studied previously. In *sequencing games* (e.g., Curiel et al. 1989), agents—each with a job that needs to be processed—start with a feasible solution on a fixed number of machines, and the profit assigned to a coalition of agents is the maximal cost savings the coalition can achieve by rearranging themselves. Scheduling games have received somewhat limited attention; some authors have developed

axiomatic characterizations of various cost sharing rules for these games (Maniquet 2003, Mishra and Rangarajan 2005).

Smith (1956) showed that scheduling jobs in nonincreasing order of w_j/p_j minimizes the sum of weighted completion times on a single machine. To simplify the analysis, for the remainder of this paper we assume without loss of generality that $w_1/p_1 \geq \dots \geq w_n/p_n$. Under this assumption, we have that $v(S) = \sum_{i \in S} \sum_{j=1, j \in S}^i w_j p_j$ for any $S \subseteq N$.

The structure of the cost function for scheduling games allows us to explicitly express a cost allocation in the least core of scheduling games and recast the least core linear program (LC) as the maximization of a set function defined solely in terms of the cost function v . We consider the cost allocation \bar{x} defined as follows:

$$\bar{x}_i = \frac{1}{2}(v(S^i) - v(S^{i-1})) + \frac{1}{2}(v(N \setminus S^{i-1}) - v(N \setminus S^i)) \quad (3)$$

$$= \frac{1}{2}w_i \sum_{j=1}^i p_j + \frac{1}{2}p_i \sum_{j=i}^n w_j \quad (4)$$

for $i = 1, \dots, n$, where $S^i = \{1, \dots, i\}$ and $S^0 = \emptyset$.

THEOREM 3. *Suppose (N, v) is a scheduling game.*

(a) *The cost allocation \bar{x} defined in (4) is in the least core of (N, v) .*

(b) *The least core value of (N, v) is $z^* = \frac{1}{2} \max_{S \subseteq N, S \neq \emptyset, N} \{v(N) - v(S) - v(N \setminus S)\}$.*

PROOF. For any $S \subseteq N$, the cost allocation \bar{x} defined in (4) satisfies

$$2(\bar{x}(S) - v(S)) = \sum_{i \in S} \sum_{j=1}^i w_i p_j + \sum_{i \in S} \sum_{j=i}^n p_i w_j - 2 \sum_{i \in S} \sum_{j=1}^i w_i p_j \quad (5a)$$

$$= \sum_{i \in S} \sum_{j=1}^i w_i p_j + \sum_{i \in S} \sum_{j=i}^n p_i w_j - \sum_{i \in S} \sum_{j=1}^i w_i p_j - \sum_{i \in S} \sum_{j=i}^n p_i w_j \quad (5b)$$

$$= \sum_{i \in S} \sum_{j=1}^i w_i p_j + \sum_{i \in S} \sum_{j=i}^n p_i w_j \quad (5c)$$

$$= \sum_{i \in S} \sum_{j=1}^i w_i p_j + \sum_{i \in N \setminus S} \sum_{j=1}^i w_i p_j \quad (5d)$$

$$= \sum_{i \in N} \sum_{j=1}^i w_i p_j - \sum_{i \in S} \sum_{j=1}^i w_i p_j - \sum_{i \in N \setminus S} \sum_{j=1}^i w_i p_j + \sum_{i \in S} \sum_{j=1}^i w_i p_j + \sum_{i \in N \setminus S} \sum_{j=1}^i w_i p_j \quad (5e)$$

$$= \sum_{i \in N} \sum_{j=1}^i w_i p_j - \sum_{i \in S} \sum_{j=1}^i w_i p_j - \sum_{i \in N \setminus S} \sum_{j=1}^i w_i p_j \quad (5f)$$

$$= v(N) - v(S) - v(N \setminus S). \quad (5g)$$

Let $\bar{z} = \frac{1}{2} \max_{S \subseteq N, S \neq \emptyset, N} \{v(N) - v(S) - v(N \setminus S)\}$. The solution (\bar{x}, \bar{z}) is feasible for (LC), since (5a)–(5g) implies $\bar{x}(N) = v(N)$, and for any $S \subseteq N, S \neq \emptyset, N$,

$$\bar{z} \geq \frac{1}{2}(v(N) - v(S) - v(N \setminus S)) = \bar{x}(S) - v(S).$$

Now suppose (x^*, z^*) is an optimal solution to (LC). As in the proof of Theorem 2, we obtain the following lower bound on $2z^*$:

$$2z^* \geq v(N) - v(S) - v(N \setminus S) \quad \text{for all } S \subseteq N, S \neq \emptyset, N.$$

Therefore, $z^* \geq \bar{z}$. It follows that the cost allocation \bar{x} is in the least core of (N, v) , and the least core value of (N, v) is \bar{z} . \square

In addition to being an element of the least core, it happens that the cost allocation \bar{x} as defined in (4) is the Shapley value of scheduling games (Mishra and Rangarajan 2005). This is quite remarkable: for an arbitrary supermodular cost cooperative game, the Shapley value is not necessarily in the least core. Example 1 illustrates this point.

One might also wonder if the cost allocation \bar{x} as defined in (3) is in the least core, or coincides with the Shapley value, for general supermodular cost cooperative games. Note that the definition of \bar{x} in (3) depends on the ordering of N (for scheduling games, we ordered N according to nonincreasing w_j/p_j). For a given permutation $\sigma: N \rightarrow N$ where $\sigma(i)$ denotes the position of player $i \in N$, we define the cost allocation \bar{x}^σ as follows:

$$\bar{x}_{\sigma^{-1}(i)}^\sigma = \frac{1}{2}(v(S^i) - v(S^{i-1})) + \frac{1}{2}(v(N \setminus S^{i-1}) - v(N \setminus S^i))$$

for $i = 1, \dots, n$, where $S^i = \{\sigma^{-1}(1), \dots, \sigma^{-1}(i)\}$, and $S^0 = \emptyset$. The cooperative game (N, v) , defined in Example 1 below, is an instance of a supermodular cost cooperative game (in particular, v is of the form (2)) for which the cost allocation \bar{x}^σ is not in the least core and is not the Shapley value, for any permutation σ of N .

EXAMPLE 1. Consider the cooperative game (N, v) defined as follows. There are four players: $N = \{1, 2, 3, 4\}$. Each agent $i \in N$ has a processing time $p_i = i$. The cost $v(S)$ to a coalition S is the minimum total completion time of jobs in S on two identical parallel machines with non-preemptive processing. By Theorem 1 and a result by Queyranne and Schulz (1995), v is supermodular. The Shapley value of this game is $\phi_1 = 3/2, \phi_2 = 17/6, \phi_3 = 23/6$, and $\phi_4 = 29/6$, and $\max_{S \subseteq N, S \neq \emptyset, N} \{\phi(S) - v(S)\} = 5/3$. However, the least core value of this game is $3/2$. It is also straightforward to check that $\max_{S \subseteq N, S \neq \emptyset, N} \{\bar{x}^\sigma(S) - v(S)\} = 2$, for all permutations σ of N .

Although computing the least core value of an arbitrary supermodular cost cooperative game is strongly NP-hard, it is still unclear at this point if this is the case for scheduling games. It turns out that computing the least core value of scheduling games is weakly NP-hard: by Theorem 3(b),

computing the least core value of scheduling games is equivalent to the problem of minimizing the sum of weighted completion times of jobs on two identical parallel machines with non-preemptive processing, which is weakly NP-hard (Bruno et al. 1974). This is in contrast to Theorem 3(a), which implies that we can compute a cost allocation in the least core of scheduling games in polynomial time.

5. Concluding Remarks

Given that the problem of computing the least core value of supermodular cost cooperative games is strongly NP-hard, one might be interested in designing approximation algorithms for this problem. In a companion paper (Schulz and Uhan 2010), we build a framework to approximate the least core value of supermodular cost cooperative games (N, v) , by approximately computing a coalition S whose *dissatisfaction* $e(x, S) = x(S) - v(S)$ is maximum, given a cost allocation x . This framework yields a $(3 + \varepsilon)$ -approximation algorithm. As a by-product, we also show how to compute accompanying approximate least core cost allocations for these games. In particular, we show how to compute a cost allocation in the *2-approximate least core* of supermodular cost cooperative games (N, v) ; that is, a cost allocation $x \in \mathbb{R}^N$ that satisfies

$$x(N) = v(N),$$

$$x(S) \leq v(S) + 2z^* \quad \text{for all } S \subseteq N, S \neq \emptyset, N,$$

where z^* is the least core value of (N, v) . Using our approximation framework, we are also able to get better performance guarantees for subclasses of supermodular cost cooperative games: we give a fully polynomial-time approximation scheme for computing the least core value of scheduling games, as well as a polynomial-time algorithm for computing the least core value and a cost allocation in the least core of a cooperative game that arises from matroid optimization.

Endnotes

1. A set function $v: 2^N \rightarrow \mathbb{R}$ is modular if (1) is satisfied with equality for all $j, k \in N$ such that $j \neq k$ and $S \subseteq N \setminus \{j, k\}$.
2. A ρ -approximation algorithm ($\rho \geq 1$) is an algorithm that always finds a solution whose objective value is within a factor ρ of the optimal value and whose running time is polynomial in the input size. The parameter ρ is known as the performance guarantee of the algorithm.
3. The Shapley value (Shapley 1953) of a cooperative game (N, v) is the cost allocation $\phi \in \mathbb{R}^N$, where

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!} (v(S \cup \{i\}) - v(S))$$

for all agents $i \in N$.

In words, the Shapley value of each agent i reflects agent i 's average marginal contribution to the coalition N . The Shapley value is a classic, well-studied solution concept in cooperative game theory; for example, see Roth (1988).

4. A set function $v: 2^N \rightarrow \mathbb{R}$ is submodular if $-v$ is supermodular.

Acknowledgments

The authors thank the associate editor, two anonymous referees, and Garrett van Ryzin for helpful feedback. This research was supported by the National Science Foundation (DMI-0426686).

References

- Bird, C. G. 1976. Cost-allocation for a spanning tree. *Networks* **6** 335–350.
- Bruno, J., E. G. Coffman, R. Sethi. 1974. Scheduling independent tasks to reduce mean finishing time. *Commun. ACM* **17** 382–387.
- Chen, X., J. Zhang. 2006. Duality approaches to economic lot sizing games. Operations Management Working Paper, OM-2006-01, Stern School of Business, New York University.
- Curiel, I., G. Pederzoli, S. Tijs. 1989. Sequencing games. *Eur. J. Oper. Res.* **40** 344–351.
- Deng, X. 1998. Combinatorial optimization and coalition games. D.-Z. Du, P. M. Pardalos, eds. *Handbook of Combinatorial Optimization*, Vol. 2. Kluwer Academic Publishers, Dordrecht, the Netherlands, 77–103.
- Edmonds, J. 1970. Submodular functions, matroids, certain polyhedra. R. Guy, H. Hanani, N. Sauer, J. Schönheim, eds. *Combinatorial Structures and Their Applications (Calgary International Conference on Combinatorial Structures and Their Applications)*. Gordon and Breach, New York, 69–87.
- Faigle, U., W. Kern, J. Kuipers. 2001. On the computation of the nucleolus of a cooperative game. *Internat. J. Game Theory* **30** 79–98.
- Faigle, U., W. Kern, D. Paulusma. 2000. Note on the computational complexity of least core concepts for min-cost spanning tree games. *Math. Methods Oper. Res.* **52** 23–38.
- Garey, M. R., D. S. Johnson, L. Stockmeyer. 1976. Some simplified NP-complete graph problems. *Theoretical Comp. Sci.* **1** 237–267.
- Gillies, D. B. 1959. Solutions to general non-zero-sum games. A. W. Tucker, R. D. Luce, eds. *Contributions to the Theory of Games, Volume IV. Annals of Mathematics Studies*, Vol. 40. Princeton University Press, Princeton, NJ, 47–85.
- Goemans, M. X., M. Skutella. 2004. Cooperative facility location games. *J. Algorithms* **50** 194–214.
- Goemans, M. X., M. Queyranne, A. S. Schulz, M. Skutella, Y. Wang. 2002. Single machine scheduling with release dates. *SIAM J. Discrete Math.* **15** 165–192.
- Granot, D., G. Huberman. 1981. Minimum cost spanning tree games. *Math. Programming* **21** 1–18.
- Hartman, B., M. Dror, M. Shaked. 2000. Cores of inventory centralization games. *Games Econom. Behav.* **31** 26–49.
- Håstad, J. 2001. Some optimal inapproximability results. *J. ACM* **48** 798–859.
- Kalai, E., E. Zemel. 1982. Totally balanced games and games of flow. *Math. Oper. Res.* **7** 476–478.
- Kern, W., D. Paulusma. 2003. Matching games: The least core and the nucleolus. *Math. Oper. Res.* **28** 294–308.
- Maniquet, F. 2003. A characterization of the Shapley value in queuing problems. *J. Econom. Theory* **109** 90–103.
- Maschler, M., B. Peleg, L. S. Shapley. 1979. Geometric properties of the kernel, nucleolus, and related solution concepts. *Math. Oper. Res.* **4** 303–338.
- Mishra, D., B. Rangarajan. 2005. Cost sharing in a job scheduling problem using the Shapley value. *Proc. 6th ACM Conf. Electronic Commerce*, ACM, New York, 232–239.

- Nemhauser, G. L., L. A. Wolsey. 1988. *Integer and Combinatorial Optimization*. Wiley, New York.
- Owen, G. 1975. On the core of linear production games. *Math. Programming* **9** 358–370.
- Potters, J., I. Curiel, S. Tijs. 1991. Traveling salesman games. *Math. Programming* **53** 199–211.
- Queyranne, M. 1993. Structure of a simple scheduling polyhedron. *Math. Programming* **58** 263–285.
- Queyranne, M., A. S. Schulz. 1995. Scheduling unit jobs with compatible release dates on parallel machines with nonstationary speeds. E. Balas, J. Clausen, eds. *Integer Programming and Combinatorial Optimization (IPCO 1995)*, *Lecture Notes in Computer Science*, Vol. 920. Springer, Berlin, 307–320.
- Roth, A. E., ed. 1988. *The Shapley Value: Essays in Honor of Lloyd S. Shapley*. Cambridge University Press, Cambridge, UK.
- Schulz, A. S., N. A. Uhan. 2010. Approximating the least core and least core value of cooperative games with supermodular costs. Working paper, Operations Research Center, MIT, Cambridge, MA.
- Shapley, L. S. 1953. A value for n -person games. H. W. Kuhn, A. W. Tucker, eds. *Contributions to the Theory of Games, Volume II. Annals of Mathematics Studies*, Vol. 28. Princeton University Press, Princeton, NJ, 307–317.
- Shapley, L. S., M. Shubik. 1966. Quasi-cores in a monetary economy with nonconvex preferences. *Econometrica* **34** 805–827.
- Smith, W. E. 1956. Various optimizers for single-stage production. *Naval Res. Logist. Quart.* **3** 59–66.
- Wolsey, L. A. 1985. Mixed integer programming formulations for production planning and scheduling problems. *Invited talk, 12th Internat. Sympos. Math. Programming, MIT, Cambridge, MA*.