Monotone cooperative games and their threshold versions

Haris Aziz Felix Brandt Paul Harrenstein
Institut für Informatik
Universität München
80538 München, Germany
{aziz,brandtf,harrenst}@tcs.ifi.lmu.de

ABSTRACT

Cooperative games provide an appropriate framework for fair and stable resource allocation in multiagent systems. This paper focusses on monotone cooperative games, a class which comprises a variety of games that have enjoyed special attention within AI, in particular, skill games, connectivity games, flow games, voting games, and matching games. Given a threshold, each monotone cooperative game naturally corresponds to a simple game. The core of a threshold version may be empty, even if that is not the case in the monotonic game itself. For each of the subclasses of monotonic games mentioned above, we conduct a computational analysis of problems concerning some relaxations of the core such as the least-core and the cost of stability. It is shown that threshold versions of monotonic games are generally at least as hard to handle computationally. We also introduce the *length* of a simple game as the size of the smallest winning coalition and study its computational complexity in various classes of simple games and its relationship with computing core-based solutions. A number of computational hardness results are contrasted with polynomial time algorithms to compute the length of threshold matching games and the cost of stability of matching games, spanning connectivity games, and simple coalitional skill games with a constant number of skills.

Categories and Subject Descriptors

F.2 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity; I.2.11 [**Distributed Artificial Intelligence**]: Multiagent Systems; J.4 [**Computer Applications**]: Social and Behavioral Sciences - Economics

General Terms

Economics, Theory and Algorithms

Keywords

Game theory, social choice and voting, cooperative games, core, least core, nucleolus, computational complexity

Cite as: Monotone cooperative games and their threshold versions, Haris Aziz, Felix Brandt, Paul Harrenstein, *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, van der Hoek, Kaminka, Lespérance, Luck and Sen (eds.), May, 10–14, 2010, Toronto, Canada, pp. XXX-XXX.

1. INTRODUCTION

Fairness and stability of coalitions are fundamental issues in multiagent systems and are typically modeled using the framework of cooperative game theory. In a cooperative game, the payoff or profit that each subset of agents can achieve by cooperating is called the *value* of a coalition. The crucial questions are which coalitions are stable and how the coalitions should divide the payoff among their members. Cooperative game theory has provided differing answers to these questions in the form of *solution concepts*, some of which are backed by appealing axiomatizations. An algorithmic and computational lens on these solutions is critical as large multiagent systems are deployed.

In this paper we examine monotone cooperative games with transferable utility where the value of a superset of a coalition is at least as large as the original coalition's value. Special focus is given to *simple cooperative games* where all values are either zero or one. Cooperative games can be unstable in the sense that the core is empty. We, therefore examine relaxations of the core such as the least core and the cost of stability.

Related Work.

The area of cooperative game theory has seen considerable growth over the last few decades [22]. Concepts from cooperative game theory have been used in various combinatorial optimization problems in operations research and multiagent systems which involve resource allocation among multiple players [7]. The core was introduced by Gillies [18] and led to the subsequent development of other solutions such as the nucleolus [25]. We also consider the recently introduced *cost of stability (CoS)* [5, 24] where a minimum payment is made by an external agent to incentivize cooperation among the players.

Although algorithms to compute different solutions have already been considered in the operations research literature, Deng and Papadimitriou [9] undertook one of the earliest investigations of solution concepts in terms of *computational complexity*. Deng and Fang [8] surveyed the developments in algorithmic cooperative game theory in a recent article.

Numerous classes of cooperative games have been the subject of recent research in multiagent systems: weighted voting games [13], skill games [3], multiple weighted voting games [11], network flow games [4], and spanning connectivity games [1]. Network flow games were first considered by Kalai and Zemel [19], who also proved that network flow games always have a non-empty core. Matching games have

been widely studied in game theory (see, e.g., [20]). In this paper, we also consider a natural variation called threshold matching games.

Contribution.

We look at the correspondence between monotone cooperative game and threshold simple game versions. Well-studied games such as threshold network flow games [4] fit into this framework. It is shown that the threshold version of a game is generally at least as hard to handle from a computational point of view.

In Section 3, we present the length of a simple game, i.e., the size of the smallest winning coalition, as an important game-theoretic property. If each player has a uniform cost of being influenced, then influencing the winning coalition with the smallest size is the most efficient way to influence the overall decision. Similarly, in robotics it is desirable to complete a task with a minimum number of robots. We characterize the complexity of computing the length of well-known simple games. Complexity of length is also conducive to a better understanding of the (computational) disparity between monotone games and their threshold versions.

In Section 4, a number of $\mathcal{NP}$-hardness results are contrasted with polynomial time algorithms to compute the length of threshold matching games and the cost of stability of matching games and spanning connectivity games. Equivalence between *simple coalitional skill games (SCSGs)* and a subclass of *multiple weighted voting games* is shown, which helps answer algorithmic questions concerning SCSGs. Computing the worst excess of a least core payoff vector of a game efficiently is unclear even if a least core payoff vector is given. We prove that an oracle to compute a least core payoff vector for a simple game in a *passer-consistent* representation can be used to compute the worst excess of a least core payoff vector.

In Section 5, we furthermore present structural results of the least core and the nucleolus payoffs of monotone cooperative games and conclude the paper with Section 6.

2. PRELIMINARIES

In this section, we define important classes of cooperative games and introduce various solution concepts and fundamental computational problems associated with these concepts.

2.1 Cooperative games

We begin with the formal definition of a *transferable utility cooperative game*.

DEFINITION 1. *A* cooperative game with transferable utility *is a pair (N, v) where $N = \{1, \ldots, n\}$ is a set of players and $v : 2^N \to \mathbb{R}^+$ is a* characteristic or valuation function *that associates with each coalition $S \subseteq N$ a payoff $v(S)$ where $v(\emptyset) = 0$.*[1] *A game (N, v) is* monotonic *if $v(S) \leq v(T)$ whenever $S \subseteq T$.*

Throughout the paper, when we refer to a cooperative game, we assume such a cooperative game with transferable utility. For the sake of brevity, we will also assume the set of players to be given and sometimes refer to the game (N, v) by v.

[1]Throughout the paper, we assume $0 \in \mathbb{R}^+$.

DEFINITION 2. *A* simple game *is a monotonic cooperative game (N, v) with $v : 2^N \to \{0, 1\}$ such that $v(\emptyset) = 0$ and $v(N) = 1$. A coalition $S \subseteq N$ is* winning *if $v(S) = 1$ and* losing *if $v(S) = 0$. A* minimal winning coalition *(MWC) of a simple game v is a winning coalition in which defection of any player makes the coalition losing. A simple voting game can be represented by (N, W^m), where W^m is the set of minimal winning coalitions.*

For any monotone cooperative game, one can construct a corresponding threshold game. Threshold versions are common in the multi-agent systems literature (see for instance [4, 13]).

DEFINITION 3. *For each cooperative game (N, v) and each threshold $t \in \mathbb{R}^+$, the* corresponding threshold game *is defined as the cooperative game (N, v^t), where*

$$v^t(S) = \begin{cases} 1 & \text{if } v(S) \geq t, \\ 0 & \text{otherwise.} \end{cases}$$

It is easily verified that, for any threshold t, if a game (N, v) is monotone, so is its threshold version (N, v^t), in which case (N, v^t) is a simple game.

2.2 Classes of Monotonic Games

We now review a number of specific classes of monotone cooperative games. Here we adopt the convention that if CLASS denotes a particular class of games, we have T-CLASS refer to the class of threshold games corresponding to games in CLASS, i.e., for every threshold t, (N, v^t) is in T-CLASS if and only if (N, v) is in CLASS.

Voting games are a widely used class of monotonic games.

DEFINITION 4. *A* weighted voting game (WVG) *is a simple game (N, v) for which there is a quota $q \in \mathbb{R}^+$ and a* weight *w_i for each player i such that*

$$v(S) = 1 \text{ if and only if } \sum_{i \in S} w_i \geq q.$$

The WVG with quota q and weights $w_1, \ldots, w_n$ for the players is also denoted by $[q; w_1, \ldots, w_n]$, where we commonly assume $w_i \geq w_{i+1}$ for $1 \leq i < n$.

A multiple weighted voting game *(MWVG) is the simple game (N, v) for which there are WVGs $(N, v_1), \ldots, (N, v_m)$ such that*

$$v(S) = 1 \text{ if and only if } v_k(S) = 1, \text{ for all } k \text{ with } 1 \leq k \leq m.$$

We also denote the MWVG game composed of $(N, v_1), \ldots, (N, v_m)$ by $(N, v_1 \wedge \cdots \wedge v_m)$.

Other important classes of games are defined on graphs. Among these are *spanning connectivity games*, *matching games*, *network flow games*, and *graph games*, where either nodes or edges are controlled by players and the value of a coalition of players depends on their ability to connect the graph, enable a bigger flow, or obtain a heavier matching or edge set.

DEFINITION 5. *For each connected undirected graph (V, E), we define the* spanning connectivity game (SCG) *as the simple game (N, v) where $N = E$ and for all $S \subseteq E$, $v(S) = 1$ if and only if there exists some $E' \subseteq S$ such that $T = (V, E')$ is a spanning tree.*

DEFINITION 6. *Let $G = (V, E, w)$ be a weighted graph. The* matching game *corresponding to G is the cooperative game (N, v) with $N = V$ and for each $S \subseteq N$, the value $v(S)$ equals the weight of the maximum weighted matching of the subgraph induced by S.*

Deng and Papadimitriou [9] introduced *graph games*, which are likewise defined on weighted graphs [9].

DEFINITION 7. *For a weighted graph (V, E, w), the* graph game (GG) *is the cooperative game (N, v) where $N = V$ and for $S \subseteq N$, $v(S)$ is the weight of edges in the subgraph induced by S. In this paper, we assume that the graph corresponding to a graph game has only positive edge weights and denote such graph games by GG^+.*

A flow network (V, E, c, s, t) consists of a directed graph (V, E), with capacity on edges $c : E \to \mathbb{R}^+$, a source vertex $s \in V$, and a sink vertex $t \in V$. A network flow is a function $f : E \to \mathbb{R}^+$, which obeys the capacity constraints and the condition that the total flow entering any vertex (other than s and t) equals the total flow leaving the vertex. The value of the flow is the maximum amount flowing out of the source.

DEFINITION 8. *For a flow network (V, E, c, s, t), the associated* network flow game (NFG) *is the cooperative game (N, v), where $N = E$ and for each $S \subseteq E$ the value $v(S)$ is the value of the maximum flow f with $f(e) = 0$ for all $e \in E \setminus S$.*

Finally, we define the class of skill games, which were recently introduced by Bachrach and Rosenschein [3]. The intuition underlying skill game is that there is a task to be performed requiring skills $\sigma_1, \ldots, \sigma_k$. Each agent has particular skills. A coalition that can perform the task achieves value 1 and a coalition that cannot perform the task gets value 0.

DEFINITION 9. *Let $N = \{1, \ldots, n\}$ and $\Sigma = \{\sigma_1, \ldots, \sigma_k\}$ be a set of players and a set of skills, respectively, such that each player has a set of skills $\Sigma_i \subseteq \Sigma$. Given a task requiring all of the skills in Σ for its performance, the* simple coalitional skill game (SCSG) *associated with it is the simple game (N, v) such that for all coalitions $S \subseteq N$,*

$$v(S) = 1 \text{ if and only if } \bigcup_{i \in S} \Sigma_i = \Sigma.$$

2.3 Cooperative Solutions

A solution for a cooperative game consists of a distribution of the values of the coalitions that form. In this paper, we assume that the grand coalition consisting of all players always forms. Accordingly, a solution consists of a distribution of the value of the grand coalition over the players. Thus, formally, a solution associates with each cooperative game (N, v) a set of *payoff vectors* $(x_1, \ldots, x_n) \in \mathbb{R}^N$ such that $\sum_{i \in N} x_i = v(N)$, where x_i denotes player i's share of $v(N)$. As such, solution concepts formalize the notions of fair and stable payoff vectors. In what follows, we use notation similar to that of Elkind et al. [13].

Given a cooperative game (N, v) and payoff vector $x = (x_1, ..., x_n)$, the *excess of a coalition S under x* is defined by

$$e(x, S) = x(S) - v(S),$$

where $x(S) = \sum_{i \in S} x_i$. We are now in a position to define one of the most fundamental solution concepts of cooperative game theory, viz., the core.

DEFINITION 10. *A payoff vector $x = (x_1, \ldots, x_n)$ is in the* core *of a cooperative game (N, v) if and only for all $S \subset N$, $e(x, S) \geq 0$.*

A core payoff vector guarantees that each coalition gets at least what it could gain on its own. The core is a desirable solution concept, but, unfortunately it is empty for many games. Games which have a non-empty core are called *balanced*. The possibility of the core being empty led to the development of the *ϵ-core* [27] and the *least core* [21].

The *excess vector* of a payoff vector x, is the vector $(e(x, S_1), ..., e(x, S_{2^n}))$ where $e(x, S_1) \leq e(x, S_2) \leq e(x, S_{2^n})$. We denote the distinct values in the excess vector by $-\epsilon_1(x, v), -\epsilon_2(x, v), \ldots, -\epsilon_m(x, v)$, where $-\epsilon_i(x, v) < -\epsilon_j(x, v)$ for $i < j$.

DEFINITION 11. *For $\epsilon > 0$, a payoff vector vector x is in the* ϵ-core *if for all $S \subset N$, $e(x, S) \geq -\epsilon$. The payoff vector x is in the* least core *if it is in the ϵ-core for the smallest ϵ for which the ϵ-core is non-empty. We will denote by $-\epsilon_1(v)$, the worst excess of any least core payoff vector of (N, v).*

It is easy to see from the definition of the least core, that it is the solution of the following linear program (LP):

$$\begin{array}{ll} \min & \epsilon \\ \text{s.t.} & x(S) \geq v(S) - \epsilon \text{ for all } S \subset N, \\ & x_i \geq 0 \text{ for all } i \in N, \\ & \sum_{i=1,\ldots,n} x_i = v(N) \ . \end{array} \tag{1}$$

The nucleolus is a special payoff vector which is in the core if the core exists and is otherwise a member of the least core.

DEFINITION 12. *A payoff vector x such that $x_i \geq v(\{i\})$ for all $i \in N$ and x has lexicographically the largest excess vector is called the* nucleolus.

The nucleolus is unique and always exists as long as $v(S) = 0$ for all singleton coalitions [25].

If the core of a coalitional game is empty, it is hard to ensure that players do not break off from the grand coalition to maximize their payoff. One recent proposal [5] to take care of this problem is the idea of an external authority paying a *supplemental payment* to incentivize the players to cooperate in a stable manner. This payment is denoted by $\triangle$. We use the same definitions as introduced by Bachrach et al. [5].

DEFINITION 13. *For a given coalitional game $G = (N, v)$ and a payment $\triangle \in \mathbb{R}^+$, the* adjusted coalitional game $G(\triangle) = (N, v')$ *is exactly like (N, v) except that $v'(N) = v(N) + \triangle$. Any payoff vector which is in the core of $G(\triangle) = (N, v')$ is a* superimputation. *The* cost of stability (CoS) *of a game is the minimum supplemental payment $CoS(G)$ such that $G(CoS(G))$ has a nonempty core.*

If the core of a game is nonempty, then, clearly, the CoS is 0. It is known that computing the CoS is $\mathcal{NP}$-hard for WVGs [5] and T-NGFs [24]. It is easy to see that $CoS(G)$ is the solution of the following LP with an exponential number of constraints:

$$\begin{array}{ll} \min & \triangle \\ \text{s.t.} & x(S) \geq v(S) \text{ for all } S \subset N \ , \\ & x_i \geq 0 \text{ for all } i \in N, \\ & \sum_{i=1,\ldots,n} x_i = v(N) + \triangle \ . \end{array} \tag{2}$$

2.4 Computational Problems

For any solution concept X, we consider the following standard computational problems:

- IN-X: given a cooperative game (N, v) and payoff vector p, check whether p is in solution X of (N, v).
- CONSTRUCT-X: given a cooperative game (N, v), compute a payoff vector p, which is in solution X of (N, v).
- CoS: given a cooperative game (N, v), compute $CoS((N, v))$.
- SUPERIMP: given cooperative game $G = (N, v)$, supplement payment $\triangle$, and super-imputation $(x_1, \ldots, x_n)$ in $G(\triangle)$, check whether $x \in CORE(G(\triangle))$.

3. LENGTH OF SIMPLE GAMES

The *length* of a simple game is the size of the smallest winning coalition. In another context, a related concept has been considered in the electrical engineering and threshold logic literature [23]. The length is an important indicator of a simple voting game which signifies the ease with which the status quo can be changed. Complexity of the length of a simple game is also related to the complexity of bribery, manipulation and control which has been an active area of research in social choice theory [14]. For example, if each player has a uniform cost of being influenced, then influencing the winning coalition with the smallest size is the most efficient way to influence the overall decision. Similarly, in a network setting, a strategic agent with limited resources would like to control the minimum number of edges or nodes which can still get the job done. As we will see in Section 4, length is an important property of simple games with a bearing on the complexity of solutions. We will denote the problem of computing the length of a game by LENGTH and the length of game (N, v) by $l(v)$.

For a simple game v represented by (N, W^m), $l(v)$ can be computed in linear time by scanning the winning coalitions and identifying the smallest k such that coalition S is in W^m and $|S| = k$. The length of an SCG is always $n-1$. Similarly, the length of a WVG can be computed in polynomial time by iteratively adding players in the order of decreasing weights until the summed weight exceeds the quota. This greedy method also identifies the smallest winning coalition.

PROPOSITION 1. *Computing the length of an MWVG is $\mathcal{NP}$-hard even if the game is composed of two WVGs.*

PROOF. We provide a reduction from a special case of the minimization version of the multidimensional 0-1 knapsack problem (MKP) [16].

Name: MIN-MKP
Instance: A collection of n items and m knapsacks, where each knapsack i should have at least d_i capacity filled and the jth item has corresponding cost c_j and requires a_{ij} units of resource consumption.
Output: Minimize $\sum_{j=1}^{n} c_j y_j$ such that $\sum_{j=1}^{n} a_{ij} y_j \geq d_i$, $i \in M = \{1, 2, \ldots m\}$ and $y_j \in \{0, 1\}$, $j \in N$.

Dinic and Karzanov [10] proved that even the special case of MIN-MKP where $m = 2$ and $c_j = 1$ for all $j =\in \{1, \ldots, n\}$ is $\mathcal{NP}$-hard. For any such restricted instance of MIN-MKP, construct a MWVG with n players and m WVGS where for $i = 1, \ldots, m$ the ith MWVG is $[d_i; a_{i1}, \ldots, a_{in}]$. Then the objective of MIN-MKP is less than x if and only if $l(v) < x$. □

PROPOSITION 2. *Computing the length of a SCSG is $\mathcal{NP}$-hard even if each skill is owned by exactly two players or if each player has three or fewer skills.*

PROOF. We provide a reduction from Vertex Cover. Consider a graph (V, E) where $V = \{1, \ldots, n\}$. Denote the neighbours of vertex i by $D(i)$. Based on the graph G, define a SCSG where $\Sigma = E$, $N = V$ where player i corresponds to vertex i and for any player i, $\Sigma_i = \{(i, j) \mid j \in D(i)\}$. A coalition S is winning if and only if $\bigcup_{i \in S} \Sigma_i = \Sigma$ which is possible if and only if the set of vertices corresponding to C is a vertex cover of (V, E).
Similarly, it can be shown that computing the length of the SCSG is equivalent to solving the Minimum Set Cover problem which is $\mathcal{NP}$-complete even if each set has size 3. □

For computing the length of threshold matching games, greedy approaches do not work and the winning coalition with the smallest cardinality need not be a subset of the maximum matching of the whole graph. Nevertheless, the length of threshold matching games can be computed in polynomial time.

PROPOSITION 3. *There exists a polynomial-time algorithm to compute the smallest winning coalition of the threshold matching game.*

PROOF. Take weighted graph $G = (V, E, w)$. Suppose we want to compute the maximum matching of size s. Then transform graph G into G' by creating $j = |V| - 2s$ new nodes $V' = \{v'_1, \ldots, v'_j\}$ and joining each node in V' to each node in V with an edge of weight $W = \sum_{i=1}^{|E|} w(e_i)$. Let M' be the maximum (perfect) matching of G'. Then $M = M' \cap E$ is the maximum matching of G with size s.
Use the procedure outlined above for $s = 1, 2, \ldots, \lfloor |V|/2 \rfloor$ and stop when $w(M' \cap E) \geq k$. Then $2s$ is the length of the game and $V(M)$ is the smallest winning coalition. □

By a reduction from the $\mathcal{NP}$-hard *Dense k-Subgraph Problem* [15], it can be shown that computing the length of threshold versions of positive graph games T-GG$^+$ is $\mathcal{NP}$-hard. Similarly, computing the length of T-NFGs is $\mathcal{NP}$-hard. This follows from an $\mathcal{NP}$-hard restricted version of the MINIMUM EDGE COST-FLOW problem [17].

DEFINITION 14. *For an unweighted graph $G = (V, E)$, the* independent set game *(ISG) is a cooperative game (N, v) where $N = E$ and for $S \subseteq E$, $v(S)$ is the size of the maximum independent set of the graph restricted to vertices of S.*

ISGs are examples of monotone cooperative games for which computing the value of a coalition is $\mathcal{NP}$-hard. Computing the length of any threshold monotone cooperative game based on an $\mathcal{NP}$-hard combinatorial optimization domain is $\mathcal{NP}$-hard. Take the example of threshold independent set games (T-ISG). If there is an oracle to compute the length of a T-ISG, then it can be used to solve the $\mathcal{NP}$-hard maximum independent set problem. Use the oracle to compute the smallest winning coalition of graph G with

Game class	Complexity of LENGTH
WVG	$\mathcal{P}$
T-Matching	$\mathcal{P}$
T-NFG	$\mathcal{NP}$-hard
MWVG	$\mathcal{NP}$-hard
SCSG	$\mathcal{NP}$-hard
T-GG$^+$	$\mathcal{NP}$-hard

Table 1: Complexity of LENGTH

gradually increasing threshold k until no winning coalition is returned. The size of the maximum independent set is then $k-1$.

Table 1 summarizes the complexity of LENGTH for different game classes. We will connect these results to the complexity of computing least core payoff vectors and the CoS in Proposition 7 and Observation 2.

4. LEAST CORE AND COS

For a given payoff vector, computing the worst excess is a non-trivial task. Deng and Fang [8] note that *"the most natural problem is how to efficiently compute the value ϵ_1 for a given cooperative game. The catch is that the computation of ϵ_1 requires one to solve a linear program with [an] exponential number of constraints."* It is not clear whether the least core worst excess can be computed efficiently even if a least core payoff vector is given. However, we show that any oracle that can be used to compute a least core payoff vector for *passer-consistent* simple games, can also be used to compute the worst excess for the least core payoff vector. A class of simple games is *passer-consistent* if, for any simple game contained in the class, the game with a newly added passer is also contained in the class. WVGs, MWVGs, and SCSGs are examples of passer-consistent representations.

Let (N,v) be a simple game and x be any payoff vector of (N,v). We will denote $1-\epsilon_1(x,v)$ by $\delta_1(x,v)$ and $1-\epsilon_1(v)$ by $\delta_1(v)$. The value $\delta_1(x,v)$ is the payoff of any coalition with the worst excess.

LEMMA 1. *Let (N,v) be a simple game and x be any payoff vector of (N,v) such that $x(N)=1$. Consider the game $(N\cup\{n+1\},v')$ which is obtained by adding a passer player $n+1$ to the game (N,v). For any payoff vector x' for $(N\cup\{n+1\},v')$, if $x'_{n+1}=a$ and $x'_i=(1-a)x_i$ for $i\in N$, then*

1. *$\delta_1(x',v')=\mathsf{Min}(a,(1-a)\delta_1(x,v))$.*
2. *If x' is a least core payoff vector of $(N\cup\{n+1\},v')$, then $a=(1-a)\delta_1(x,v)$ and $x'_{n+1}(v')=\delta_1(v')$.*

PROOF. Since $\{n+1\}$ is a winning coalition, the worst excess of $\{n+1\}$ for payoff vector x' is $a-1$ which implies that $\delta_1(x',v')\leq a$.

For $S\subseteq N$, any coalition $S\cup\{n+1\}$ is not a minimal winning coalition. Therefore, to examine other coalitions with the worst excess in $(N\cup\{n+1\},v')$ for payoff vector x', we look at subsets of N. The worst payoff for winning coalitions among N is then $(1-a)\delta_1(x,v)$. This implies that $\delta_1(x',v')\leq(1-a)\delta_1(x,v)$. Since all subsets of $N\cup\{n+1\}$ have been considered, $\delta_1(x',v')=\mathsf{Min}(a,(1-a)\delta_1(x,v))$.

We now prove that payoff vector x' is a least core payoff vector of $(N\cup\{n+1\},v')$, only if $a=(1-a)\delta_1(x,v)$ and $x'_{n+1}(v')=\delta_1(v')$.

The value $\delta_1(x',v')$ is maximized only when $a=(1-a)\delta_1(x,v)$. Also, $\delta_1(x',v')$ is maximum only when the optimum payoff $\delta_1(v')$ is given to player $n+1$, i.e., when $x'_{n+1}=a=\delta_1(v')$. □

PROPOSITION 4. *An oracle to compute a least core payoff vector for a simple game in any passer-consistent representation can be used to compute the worst excess of a least core payoff vector.*

PROOF. Consider a game (N,v) in a passer-consistent representation. Use the oracle to compute $x=(x_1,\ldots,x_n)$, a least core payoff vector of (N,v).

Denote the worst excess (as yet unknown) of x by $-\epsilon_1(v)$ and $1-\epsilon_1(v)$ by $\delta_1(v)$. Then, we know that $\delta_1(x,v)=\delta_1(v)$. Form a new game $(N\cup\{n+1\},v')$ by adding a passer player $n+1$ to the game such that $v'(\{n+1\})=1$ and $v'(S)=1$ if and and only if $v(S)=1$ for all $S\subseteq N$. Since (N,v) is in a passer-consistent representation, $(N\cup\{n+1\},v')$ can also be represented by a passer-consistent representation.

Use the oracle to compute $x'=(x'_1,\ldots x'_{n+1})$, a least core payoff vector of $(N\cup\{n+1\},v')$. From Lemma 1, we know that $x'_{n+1}(v')=\delta_1(v')$ and $x'_{n+1}=(1-x'_{n+1})\delta_1(x,v))$. This means that,

$$1-\epsilon_1(v)=\delta_1(v)=\frac{x'_{n+1}(v')}{1-x'_{n+1}(v')} \tag{3}$$

From (3), we know that $\epsilon_1(v)=1-\delta_1(v)$ can be computed by adding a passer to (N,v) to form game $(N\cup\{n+1\},v')$ and then computing $x'_{n+1}(v')$. □

Bachrach et al. [6] focus on the CoS of WVGs and note that *"a natural research direction is to study the cost of stability in other classes of games."* We now present algorithms to compute the CoS of SCGs and general matching games.

PROPOSITION 5. *For SCGs, there exists a polynomial-time algorithm to compute SUPERIMP and CoS.*

PROOF. We consider a superimputation $(x_1,\ldots,x_{|E|})$ such that $x(E)=1+\triangle$. For a candidate solution $x=(x_1,\ldots,x_{|E|})$, we find in polynomial time the minimum spanning tree T of the graph G^x. If $x(T)\geq 1$, then $x(S)\geq 1$ for all $S\subset E$ and x is a superimputation. If $x(T)<1$, then x is not a superimputation.

The size of the linear program (2) is exponential in the size of the graph G, with an inequality for every subset of edges. However, this linear program can be solved using the ellipsoid method and a separation oracle, which verifies in polynomial time whether a solution is feasible or returns a violated constraint [26]. We see that our solution to SUPERIMP for an SCG provides a separation oracle to compute CoS for the same SCG. □

PROPOSITION 6. *For matching games, there exists a polynomial-time algorithm to compute SUPERIMP and CoS.*

PROOF. We denote weighted graph $G=(N,E,w)$ and its corresponding matching game also by G. The CoS LP can be solved using the ellipsoid method and a polynomial time separation oracle which for any payoff vector $x=(x_1,\ldots,x_n)$ and $\epsilon>0$, returns "yes" if the worst excess of G with respect to x is more than $-\epsilon$ and otherwise returns the violated constraint. We construct the

separation oracle as follows. For a payoff vector x and $G = (N, E, w)$, the graph G'_x is (N, E, w'), where for each edge (i, j), $w'((i, j)) = w((i, j)) - x_i - x_j$. For any coalition S, $-e(x, S)$ is equal to the weight of a maximum matching of G'_x restricted to nodes in S. Therefore, it is easy to see that for matching game G, if the weight of the maximum matching of G'_x is a positive value ϵ_1, then, the worst excess of x is $-\epsilon_1$. Since a maximum weighted matching can be computed in polynomial time, one can compute the worst excess $-\epsilon_1$ for payoff vector x and if $\epsilon_1 > \epsilon$, then the maximum matching of G'_x provides the violated constraint. The separation oracle can also be used to solve SUPERIMP. □

Resnick et al. [24] mention the relation between the CoS and the least core as an interesting question. The following observation highlights the similarity of the computational approach to compute both solutions. The observation stems from the the similarity of LP (1) for the least core and LP (2).

Observation 1. *For a monotone cooperative game (N, v), if the separation oracle O for a least core LP can be constructed and be solved in polynomial time, then for (N, v), SUPERIMP and CoS are in P. The reasoning for this is as follows. Consider payoff vector $(x_1, \ldots, x_n)$. Then, oracle O can check in polynomial time whether $x(S) - v(S) \geq -\epsilon$ for all $S \subset N$, or find a violated constraint otherwise. Then O can be used to solve SUPERIMP for (N, v). Also, O can be used as a separation oracle to solve LP (2).*

Proposition 7. *If computing the length of a simple game (N, v) is $\mathcal{NP}$-hard, then IN-ϵ-CORE for (N, v) is $\mathcal{NP}$-hard.*

Proof. Consider the payoff vector $x = (\frac{1}{n}, \ldots, \frac{1}{n})$ for (N, v). Denote the length of (N, v) by $l(v)$. The payoff of the smallest winning coalition is $\frac{l(v)}{n}$. The worst excess of (N, v) for payoff vector x is $\frac{l(v)}{n} - 1$.

The payoff vector x is in the ϵ-core if and only if $\frac{l(v)}{n} \geq 1 - \epsilon$. If there is an oracle to compute IN-ϵ-CORE in polynomial time, then by using different values of ϵ, binary search can be used to compute the $l(v)$. Therefore computing $l(v)$ reduces to solving IN-ϵ-CORE. Since $l(v)$ is $\mathcal{NP}$-hard to compute, IN-ϵ-CORE is $\mathcal{NP}$-hard. □

One would expect that if IN-CORE is in $\mathcal{P}$ then SUPERIMP is in $\mathcal{P}$. This is because for each coalition $S \subset N$, we need to check if $x(S) \geq v(S)$. However, for WVGs, IN-CORE is in $\mathcal{P}$ but SUPERIMP is co$\mathcal{NP}$-complete. Moreover, due to the supplemental payment, the game does not remain a simple game anymore.

Observation 2. *If IN-ϵ-CORE is $\mathcal{NP}$-hard and unless $\mathcal{P} = \mathcal{NP}$, then there is no polynomial time separation oracle to solve the least core LP or the CoS LP. The reason is as follows. A separation oracle for solving the least core LP is a polynomial time algorithm which verifies in polynomial time whether a solution is feasible or returns a violated constraint [26]. Such an oracle can be used to solve IN-ϵ-LEAST-CORE which only requires a yes/no output and not the violated constraint.*

As we saw in Section 3, for many important simple coalitional games such as SCSGs, computing the length is $\mathcal{NP}$-hard. Proposition 7 and Observation 2 imply that if computing the length of a simple game is $\mathcal{NP}$-hard and unless $\mathcal{P} = \mathcal{NP}$, then there is no polynomial time separation oracle to solve the least core LP. This means that, if a polynomial time algorithm does exist, one needs to make extra use of the combinatorial structure of the underlying game. For the threshold matching games, although the length can be computed in polynomial time, solving IN-ϵ-CORE is $\mathcal{NP}$-hard.

Proposition 8. *For threshold matching games, IN-ϵ-CORE is $\mathcal{NP}$-hard.*

Proof. Denote by WORST-EXCESS the problem of computing the worst excess of a cooperative game with respect to a given payoff vector x. We provide a reduction from the $\mathcal{NP}$-hard Minimization Knapsack Problem (MinKP) to WORST-EXCESS for threshold matching games. Recall, that MinKP is the problem of minimizing $\sum_{i=1}^{n} p_i y_i$ such that $\sum w_i y_i \geq d$ where $y_i \in \{0, 1\}$ for $i = 1, \ldots, n$.

Let $P = \sum_{i=1}^{n} p_i$. For an instance I of MinKP, denote the solution by s^*. For I, we can construct an instance of WORST-EXCESS for threshold matching game G_I. Game G_I consists of threshold d and weighted graph (V, E, w) where $|V| = 2n$, $|E| = n$ and $E = \{(v_{2i-1}, v_{2i}) \mid i = 1, \ldots, n\}$ such that $w((v_{2i-1}, v_{2i})) = w_i$. The payoff vector $x = (x_1, \ldots, x_{2n})$ for vertices $v_1, \ldots, v_{2n}$ is $x_{2i-1} = x_{2i} = p_i/2P$ for all $i \in N$. Let $-\epsilon_1$ be the worst excess of matching game G_I with respect to x. Then $s^* = (1 - \epsilon_1)P$. Therefore WORST-EXCESS is $\mathcal{NP}$-hard for threshold matching games. Since an oracle to solve IN-ϵ-CORE can be used to solve WORST-EXCESS, we see that IN-ϵ-CORE is $\mathcal{NP}$-hard for threshold matching games. □

Proposition 9. *Computing the cooperative game solutions of threshold matching games is at least as hard as computing them for WVGs.*

Proof. Take any WVG $v_I = [q; w_1, \ldots, w_n]$. Create a corresponding threshold matching game G_I. Game G_I consists of threshold q and weighted graph (V, E, w) where $|V| = 2n$, $|E| = n$ and $E = \{(v_{2i-1}, v_{2i}) \mid i = 1, \ldots n\}$ such that $w((v_{2i-1}, v_{2i})) = w_i$. Then if there an oracle to compute solution $\mathcal{S}$, for G_I in polynomial time, then it can be used to compute $\mathcal{S}$ for v_I in polynomial time. □

As a corollary, for threshold matching games, computing the cooperative solutions Shapley values and Banzhaf values is #P-complete and computing the nucleolus, the CoS, or an element in the least core is $\mathcal{NP}$-hard.

By using the same reduction, Proposition 9 also applies to the threshold version of graph games (T-GG$^+$). Proposition 9 shows that threshold matching games are harder to deal with than WVG but easier to handle than MWVGs for which even computing the length is $\mathcal{NP}$-hard.

Proposition 10. *A SCSG with n players and k skills is equivalent to a MWVG with n players and k constituent WVGs, each with quota one and weights zero or one.*

Proof. Consider SCSG (N, v) with n players and k skills. Then for $j = 1, \ldots, k$ and for each skill σ_j, construct a corresponding WVG $(N, v_j) = [q^j; w_1^j, \ldots, w_n^j]$ where $q^j = 1$ and for $i = 1, \ldots, n$, $w_i^j = 1$ if i has skill s_j and zero otherwise. Denote the MWVG $(N, v_1 \wedge \cdots \wedge v_k)$ by (N, v'). Then for any coalition $S \subseteq N$, $v(S) = 1$ if and only if $v'(S) = 1$. The same idea is used to construct a reduction in the opposite direction. □

COROLLARY 1. *For a SCSG with a constant number of skills, both computing the CoS and computing the nucleolus is in $\mathcal{P}$.*

PROOF. Elkind and Pasechnik [12] outlined an algorithm for computing the nucleolus of a MWVG which is polynomial in n and the sum of the weights of the WVGs. A SCSG with a fixed number of skills can be reduced to its corresponding MWVG. This MWVG clearly has weights polynomial in n. It follows that computing the nucleolus (which is a least core payoff vector) of SCSGs with a constant number of skills is in $\mathcal{P}$. From Observation 1, we know that the separation oracle for the least core linear program $\mathcal{LP}^1$ (defined by Elkind and Pasechnik [12]) can be used for the CoS LP. □

5. STRUCTURE OF LEAST CORE PAYOFF VECTORS

In this section, we present some general structural results which apply to least core payoff vectors of monotone cooperative game or in other cases to general simple games. Such results shed more light on stability-based solutions of monotone cooperative games.

PROPOSITION 11. *For any monotone cooperative game (N, v), suppose that $x = (x_1, \ldots, x_n)$ is an element in the least core, where the minimum excess is $-\epsilon$. Then for any player $i \in N$ there exists a coalition T such that $i \in T$ and $e(x, T) = -\epsilon$.*

PROOF. Let A be the set of players such that for every $j \in A$, we have that j is contained in some coalition M with $e(x, M) = -\epsilon$. Let P be the set of those coalitions which get an excess of $-\epsilon$. Consider a player $i \in N \setminus A$.

Let $x_i = 0$. If coalition $S \in P$, then we consider the coalition $S \cup \{i\}$. If $v(S \cup \{i\}) = v(S)$, then $i \in A$. If $v(S \cup \{i\}) > v(S)$, then $e(x, S \cup \{i\}) < -\epsilon$, which is a contradiction.

Now consider the case when $x_i > 0$. Let δ be half of the minimum of the non-zero differences between successive components of the excess vector of x. If there exists a coalition $S \in P$ such that $x(S \cup \{i\}) - v(S \cup \{i\}) < -\epsilon$, then this is a contradiction. If there exists a coalition $S \in P$ such that $x(S \cup \{i\}) - v(S \cup \{i\}) = -\epsilon$, then $i \in A$. If there exists no coalition $S \in P$ such that $x(S \cup \{i\}) - v(S \cup \{i\}) \leq -\epsilon$, then we can obtain a new payoff vector y such that $y_i = x_i - \mathsf{Min}(x_i, \delta)$, and $y_j = x_j + \frac{\mathsf{Min}(x_i, \delta)}{|A|}$ for $j \in A$, and $y_k = x_k$ for $k \notin A \cup \{i\}$. Since the smallest excess for y is more than $-\epsilon$, this means that x is not in the least core which is a contradiction. □

A player i in a simple game (N, v) is a *vetoer* if $v(N \setminus \{i\}) = 0$.

PROPOSITION 12. *Let (N, v) be a simple game with no vetoers and let $x = (x_1, \ldots, x_n)$ be a member of the least core of (N, v). Then, there is no player which is present in every coalition which gives the minimum excess for imputation x.*

PROOF. Let P be the set of coalitions which get the minimum excess $-\epsilon$. We already know that every player is a member in at least one element of P. Let $\delta = \min_{1 \leq i < m}((\epsilon_{i+1} - \epsilon_i)/2)$ and assume there is a player j which is a member of each coalition in P. Then there are three possibilities:

1. *There exists a player i other than j such that $x_i > 0$ and i is not in every member of P.* Since j features in all coalitions in P, then player a i other than j such that $x_i > 0$ can donate $\frac{\delta}{n}$ weight to j which increases the payoffs of all coalitions in P which do not include i. This is a contradiction as x is a least core payoff vector.
2. *Any player i other than j such that $x_i > 0$ is in every member of P.* Let the set of such players be J'. Then we prove that j is a vetoer which is equivalent to saying that $v(N \setminus \{j\}) = 0$. For a contradiction, assume that $v(N \setminus \{j\}) = 1$. Then $x(N \setminus \{j\}) = x(J')$. Since we have that $J' \subseteq S$ for all $S \in P$ and since $v(N \setminus \{j\}) = 1$, we know that $N \setminus \{j\} \in P$. This is a contradiction as j is in each coalition in P.
3. *There exists no player i other than j such that $x_i > 0$.* But if this happens, $x_j = 1$. This implies $x(N \setminus \{j\}) = 0$. Also $v(N \setminus \{j\}) = 0$ because if $v(N \setminus \{j\}) = 1$, then $N \setminus \{j\}$ has the minimum possible excess but does not include j. Therefore, there exists a coalition $N \setminus \{j\}$ which also gets the worst excess (0 in this case). □

We call a payoff vector *nucleolus-like* if it is a least core payoff vector for which the number of coalitions with the worst excess is the minimum possible. For a game (N, v) and a payoff vector vector x, the set of coalitions that get the ith distinct worst excess $-\epsilon_i(x, v)$ will be denoted by $A^i_x(v)$.

PROPOSITION 13. *For any monotone cooperative game (N, v) and nucleolus-like payoff vector x, assume that there exists a player i such that for all $S \in A^1_x(v)$, we have that $i \in S$. Then, for player j other than i, either for all $S \in A^1_x(v)$, $j \in S$, or we have that $x_j = 0$.*

PROOF. Assume that there exists a coalition $S \in A^1_x(v)$ such that player $j \notin S$. Let $\delta = (\epsilon_1 - \epsilon_2)/2$. Then if j donates δ amount of its payoff to i, this reduces the number of $-\epsilon_1$-coalitions. This cannot be since x is nucleolus-like. □

6. CONCLUSIONS AND DISCUSSION

In this paper, we consider a range of monotonic cooperative games. The paper also examines the length of simple games, which is the size of the smallest winning coalition. We find that the complexity of computing the length depends on the representation.

It is seen that threshold versions of cooperative games are generally less stable and computationally harder to handle. For example, NFGs are balanced whereas a T-NFG has an empty core if it contains no vetoers. Also, computing the CoS of matching games is in $\mathcal{P}$ but computing the CoS of threshold matching games is $\mathcal{NP}$-hard. Tables 1 and 2 show that the following classes of games generally get increasingly harder to handle computationally: GG^+, NFGs, matching games, WVGS, threshold matching games, T-GG^+, T-NFG, MWVGs. The complexity of computing the CoS of SCSG with a variable number of skills is open.

There may be multiple ways of distributing the payoffs after the cost of stability has been paid. Our reflections lead us to propose a natural and desirable solution for any cooperative game called the *super-nucleolus*. The super-nucleolus is the nucleolus of a cooperative game G if the core is nonempty and is the nucleolus of $G(CoS(G))$ if the core of G is empty. For certain balanced subclasses of games such as assignment games (matching games on bipartite graphs), computing the

	least core	CoS	nucleolus
GG^+	$\mathcal{P}$[9]	$\mathcal{P}$[9]	$\mathcal{P}$[9]
SCG	$\mathcal{P}$	$\mathcal{P}$	$\mathcal{P}$ [2]
SCSG (fixed #skills)	$\mathcal{P}$	$\mathcal{P}$	$\mathcal{P}$
NFG	$\mathcal{P}$ [19]	$\mathcal{P}$[19]	?
Matching	$\mathcal{P}$ [20]	$\mathcal{P}$	?
WVG	$\mathcal{NP}$-hard [13]	$\mathcal{NP}$-hard [13]	$\mathcal{NP}$-hard [13]
T-Matching	$\mathcal{NP}$-hard	$\mathcal{NP}$-hard	$\mathcal{NP}$-hard
T-NFG	$\mathcal{NP}$-hard [24]	$\mathcal{NP}$-hard [24]	$\mathcal{NP}$-hard
$T\text{-}GG^+$	$\mathcal{NP}$-hard	$\mathcal{NP}$-hard	$\mathcal{NP}$-hard

Table 2: Summary of results

nucleolus is easier than for more general games. Since the core of $G(CoS(G))$ is always nonempty, it will be interesting to compare the complexity of computing the nucleolus of $G(CoS(G))$ and that of G.

It will be interesting to check whether computing the CoS and computing a least core payoff vector have the same complexity for any game. Longstanding open problems include the complexity of computing the nucleolus of general NFGs and matching games.

ACKNOWLEDGEMENTS

This material is based upon work supported by the Deutsche Forschungsgemeinschaft under grants BR-2312/6-1 (within the European Science Foundation's EUROCORES program LogICCC) and BR 2312/3-2. We would like to thank Mike Paterson, Oded Lachish, Rahul Savani, and Felix Fischer for useful discussions and the anonymous referees for valuable feedback.

REFERENCES

[1] H. Aziz, O. Lachish, M. Paterson, and R. Savani. Power indices in spanning connectivity games. In *Proc. of International Conference on Algorithmic Aspects in Information and Management (AAIM)*, pages 55–67, 2009.

[2] H. Aziz, O. Lachish, M. Paterson, and R. Savani. Wiretapping a hidden network. In *Proceedings of the 5th International Workshop on Internet and Network Economics (WINE)*, pages 438–446, 2009.

[3] Y. Bachrach and J. S. Rosenschein. Coalitional skill games. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 1023–1030, 2008.

[4] Y. Bachrach and J. S. Rosenschein. Power in threshold network flow games. *Journal of Autonomous Agents and Multi-Agent Systems*, 18(1):106–132, 2009.

[5] Y. Bachrach, R. Meir, M. Zuckerman, J. Rothe, and J. S. Rosenschein. The cost of stability in weighted voting games. In *Proceedings of the 8th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*.

[6] Y. Bachrach, E. Elkind, R. Meir, D. Pasechnik, M. Zuckerman, J. Rothe, and J. S. Rosenschein. The cost of stability and its application to weighted voting games. In *Proceedings of the 2nd International Symposium on Algorithmic Game Theory (SAGT)*, 2009.

[7] P. Borm, H. Hamers, and R. Hendrickx. Operations research games: A survey. *TOP*, 9(2):139–199, 2001.

[8] X. Deng and Q. Fang. Algorithmic cooperative game theory. In *Pareto Optimality, Game Theory And Equilibria*, volume 17 of *Springer Optimization and Its Applications*, pages 159–185. Springer, 2008.

[9] X. Deng and C. H. Papadimitriou. On the complexity of cooperative solution concepts. *Math. Oper. Res.*, 19 (2):257–266, 1994.

[10] E. Dinic and A. Karzanov. Boolean optimization problems with uniform constraints. *Institute for Control Sci.*, 1978. Reprint.

[11] P. E. Dunne, W. van der Hoek, S. Kraus, and M. Wooldridge. Cooperative boolean games. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 1015–1022, 2008.

[12] E. Elkind and D. Pasechnik. Computing the nucleolus of weighted voting games. In *Proceedings of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 327–335, 2009.

[13] E. Elkind, L. A. Goldberg, P. W. Goldberg, and M. J. Wooldridge. Computational complexity of weighted threshold games. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence (AAAI)*, pages 718–723. AAAI Press, 2007.

[14] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. A richer understanding of the complexity of election systems. In S. Ravi and S. Shukla, editors, *Fundamental Problems in Computing: Essays in Honor of Professor Daniel J. Rosenkrantz*. Springer-Verlag, 2009.

[15] U. Feige, D. Peleg, and G. Kortsarz. The dense k-subgraph problem. *Algorithmica*, 29(3):410–421, 2001.

[16] A. Freville. The multidimensional 0-1 knapsack problem: An overview. *European Journal of Operational Research*, 127(1):1–21, 2004.

[17] M. Garey and D. Johnson. *Computers and Intractability, a Guide to the Theory of NP-completeness*. Freeman, 1979.

[18] D. B. Gillies. Solutions to general non-zero-sum games. In A. W. Tucker and R. D. Luce, editors, *Contributions to the Theory of Games IV*, volume 40 of *(Annals of Mathematics Studies)*, pages 47–85. Princeton University Press, 1959.

[19] E. Kalai and E. Zemel. On totally balanced games and games of flow. Discussion Papers 413, Northwestern University, Center for Mathematical Studies in Economics and Management Science, 1980.

[20] W. Kern and D. Paulusma. Matching games: the least core and the nucleolus. *Math. Oper. Res.*, 28(2):294–308, 2003.

[21] M. Maschler, B. Peleg, and L. S. Shapley. Geometric properties of the kernel, nucleolus, and related solution concepts. *Math. Oper. Res.*, 4:303–338, 1979.

[22] B. Peleg and P. Sudholter. *Introduction to the Theory of Cooperative Games*. Kluwer Academic, Boston, first edition edition, 2003.

[23] K. Ramamurthy. *Coherent structures and simple games (Theory and Decision Library)*. Kluwer Academic Publishers, Netherlands, first edition edition, 1990.

[24] E. Resnick, Y. Bachrach, R. Meir, and J. S. Rosenschein. The cost of stability in network flow games. In *Proceedings of the 34th International Symposium on Mathematical Foundations of Computer Science*, 2009.

[25] D. Schmeidler. The nucleolus of a characteristic function game. *SIAM J. Appl. Math.*, 17(6):1163–1170, 1969.

[26] A. Schrijver. *Combinatorial Optimization : Polyhedra and Efficiency (Algorithms and Combinatorics)*. Springer, 2004.

[27] L. S. Shapley and M. Shubik. Quasi-cores in a monetary economy with non-convex preferences. *Econometrica*, 34:805–827, 1966.