Computational Complexity of Weighted Threshold Games

Edith Elkind
Electronics & Computer Science
University of Southampton
Southampton SO17 1BJ, UK

Leslie Ann Goldberg
Computer Science
University of Liverpool
Liverpool L69 3BX, UK

Paul Goldberg
Computer Science
University of Liverpool
Liverpool L69 3BX, UK

Michael Wooldridge
Computer Science
University of Liverpool
Liverpool L69 3BX, UK

Abstract

Weighted threshold games are coalitional games in which each player has a weight (intuitively corresponding to its voting power), and a coalition is successful if the sum of its weights exceeds a given threshold. Key questions in coalitional games include finding coalitions that are stable (in the sense that no member of the coalition has any rational incentive to leave it), and finding a division of payoffs to coalition members (an imputation) that is fair. We investigate the computational complexity of such questions for weighted threshold games. We study the *core*, the *least core*, and the *nucleolus*, distinguishing those problems that are polynomial-time computable from those that are NP-hard, and providing pseudopolynomial and approximation algorithms for the NP-hard problems.

Introduction

Coalitional games provide a simple but rich mathematical framework within which issues related to cooperation in multi-agent systems can be investigated (Deng & Papadimitriou 1994; Ieong & Shoham 2005; Conitzer & Sandholm 2006). Crudely, a coalitional game can be understood as a game in which players can benefit from cooperation. The key questions in such games relate to *which coalitions will form*, and *how the benefits of cooperation will be shared*. With respect to the former question, solution concepts such as the core have been formulated, in an attempt to characterise "stable" coalitions. With respect to the latter question, the Shapley value is perhaps the best-known attempt to characterise a fair distribution of coalitional value.

From a computational perspective, the key issues relating to coalitional games are, first, how such games should be *represented*, (since the obvious representation is exponentially large in the number of players, and is hence infeasible); and second, the extent to which cooperative solution concepts can be *efficiently computed*.

In this paper we consider the computational complexity of solution concepts for *weighted threshold games*. A weighted threshold game is one in which each player is given a numeric weight, and a coalition takes the value 1 if the sum of its weights exceeds a particular threshold, and the value 0 otherwise. Weighted threshold games are widely used in practice. For example, the voting system of the European Union is a combination of weighted threshold games (Bilbao *et al.* 2002).

From previous research, we know that for weighted threshold games it is #P-hard to compute the Shapley value of a given player, and that it is NP-hard to determine whether this value is zero (Matsui & Matsui 2001; Deng & Papadimitriou 1994; Prasad & Kelly 1990). It is also known that there is a pseudopolynomial time algorithm for computing the Shapley value by dynamic programming (Garey & Johnson 1979; Matsui & Matsui 2000). However, even approximating the Shapley value within a constant factor is intractable unless P=NP — see Remark 11. In this paper, we focus on three solution concepts — the *nucleolus*, the *core*, and its natural relaxation, the *least core*. Although the complexity of determining non-emptiness of the core has been studied for a variety of representations, comparatively little research has considered the least core and the nucleolus. Following a brief statement of the relevant definitions from coalitional game theory, we show that the problem of determining whether the core is empty is solvable in polynomial time, and that the nucleolus can be computed in polynomial time when the core is non-empty. Next, we show that it is NP-hard to construct an imputation in the least core of a weighted threshold game, or to determine whether a given imputation is in the least core, or to determine, for a given ϵ, whether the least core is the ϵ-core. We mitigate these hardness results by giving a fully polynomial-time approximation scheme for the least core. Furthermore, we show that all three problems can be solved in *pseudopolynomial time*: that is, the problems can be solved in polynomial time for weighted threshold games in which the weights are at most polynomially large in the number of players. We then show that it is NP-hard to determine whether the nucleolus payoff of a given agent is 0, which implies that it is NP-hard to compute the nucleolus payment of an agent, or to approximate this nucleolus payment within a constant factor. Nevertheless, we show that, for a wide class of weighted threshold games, it is possible to easily approximate the nucleolus payment of a minimal winning coalition.

Throughout the paper, we assume some familiarity with computational complexity (Papadimitriou 1994) and approximation algorithms (Ausiello *et al.* 1999).

Preliminary Definitions

We assume numbers are rationals, and unless explicitly stated otherwise (specifically, in Theorem 6), we assume that rational values are represented in binary. (Our results extend straightforwardly to any "sensible" representation of real numbers, but we use rationals to avoid tangential representational issues.) This allows us to use the machinery of polynomial-time reductions and NP-hardness. In all of our proofs, "polynomial" means "polynomial in the size of the input". Some of the problems we consider are *function problems*, rather than decision problems (Papadimitriou 1994, Chapter 10). We use the standard notion of NP-hardness for function computation: when we say it is NP-hard to compute a function, we mean that the existence of a polynomial-time algorithm for computing the function would imply P=NP.

We briefly review relevant definitions from coalitional game theory (Osborne & Rubinstein 1994, pp.255–298). A (rational valued) *coalitional game* consists of a set I of players, or agents, and a total function $\nu : 2^I \mapsto \mathbf{Q}$, which assigns a rational value to every coalition (subset of the agents). Intuitively, $\nu(S)$ is the value that could be obtained by coalition $S \subseteq I$ if they chose to cooperate, or form a coalition. The question of *how* the agents cooperate to obtain this value is not modeled at this level of analysis, and the question of how this value is *divided* amongst coalition members is similarly ignored for now. The *grand coalition* is the set I of all agents. Often, the value of a coalition is enhanced by the addition of a new participant, so the value of the grand coalition is maximum amongst coalition values. By rescaling, we may assume this value is 1.

An *imputation* is a division of this unit of value amongst the agents. The goal is typically to find an imputation which is "fair" in the sense that agents which contribute more to the grand coalition receive a larger share of the value of the coalition. There are many ways to formalise the notion of fairness. These are known as *solution concepts*. In this paper, we study three solution concepts: the *core*, the *least core*, and the *nucleolus*.

A *weighted threshold game* is a coalitional game G given by a set of agents $I = \{1, \dots, n\}$, their non-negative *weights* $\mathbf{w} = \{w_1, \dots, w_n\}$, and a *threshold* T; we write $G = (I; \mathbf{w}; T)$. For a coalition $S \subseteq I$, its value $\nu(S)$ is 1 if $\sum_{i \in S} w_i \geq T$; otherwise, $\nu(S) = 0$. Without loss of generality, we assume that the value of the grand coalition $\{1, \dots, n\}$ is 1. That is, $\sum_{i \in I} w_i \geq T$.

For a weighted threshold game, an *imputation* is a vector of non-negative rational numbers $(p_1, \dots, p_n)$, one for each agent in I, such that $\sum_{i \in I} p_i = 1$. We refer to p_i as the *payoff* of agent i. We write $w(S)$ to denote $\sum_{i \in S} w_i$. Similarly, $p(S)$ denotes $\sum_{i \in S} p_i$.

Given an imputation $\mathbf{p} = (p_1, \dots, p_n)$, the *excess* $e(\mathbf{p}, S)$ of a coalition S under $\mathbf{p}$ is defined as $p(S) - \nu(S)$. The *core* is a set of imputations defined as follows. An imputation $\mathbf{p}$ is in the *core* if it is the case that for every $S \subseteq I$, $e(\mathbf{p}, S) \geq 0$. Informally, $\mathbf{p}$ is in the core if it is the case that no coalition can improve its payoff by breaking away from the grand coalition because its payoff $p(S)$ according to the imputation is at least as high as the value $\nu(S)$ that it would get by breaking away.

The *excess vector* of an imputation $\mathbf{p}$ is the vector $(e(\mathbf{p}, S_1), \dots, e(\mathbf{p}, S_{2^n}))$, where $S_1, \dots, S_{2^n}$ is a list of all subsets of I ordered so that $e(\mathbf{p}, S_1) \leq e(\mathbf{p}, S_2) \leq \dots \leq e(\mathbf{p}, S_{2^n})$. In other words, the excess vector lists the excesses of all coalitions from the smallest (which may be negative) to the largest. The *nucleolus* is the imputation $\mathbf{x} = (x_1, \dots, x_n)$ that has the lexicographically largest excess vector. Intuitively, the nucleolus is a good imputation because it balances the excesses of the coalitions, making them as equal as possible. It is easy to see that the nucleolus is in the core whenever the core is non-empty. Furthermore, the core is non-empty if and only if $x_1 \geq 0$.

A natural relaxation of the notion of the core is the *least core*. We say that an imputation $\mathbf{p}$ is in the ϵ-core if $e(\mathbf{p}, S) \geq -\epsilon$ for all $S \subseteq I$; it is in the least core, if it is in the ϵ-core for some $\epsilon \geq 0$ and the ϵ'-core is empty for any $\epsilon' < \epsilon$. Clearly, the least core is always non-empty and contains the nucleolus.

Another solution concept for a coalitional game is the *Shapley value*. It is the imputation $\mathbf{p}$, $p_i = \phi(i)$, with

$$\phi(i) = \sum_{S \,:\, i \in S \subseteq I} \frac{(|I| - |S|)!(|S| - 1)!}{|I|!} (\nu(S) - \nu(S \setminus \{i\})).$$

The Core and the Least Core

We start by considering the core — perhaps the best known and most-studied solution concept in coalitional game theory. Intuitively, the core of a coalitional game contains imputations such that no sub-coalition could obtain a better imputation for themselves by defecting from the grand coalition. Asking whether the grand coalition is stable thus amounts to asking whether the core of the game is non-empty.

Name EMPTYCORE.
Instance Weighted threshold game $(I; \mathbf{w}; T)$.
Question Is the core empty?

The following theorem shows that EMPTYCORE is solvable in polynomial time, and that computing the nucleolus can be done in polynomial time when the core is non-empty.

Theorem 1. *The core of a weighted threshold game $G = (I, \{w_1, \dots, w_n\}, T)$ is non-empty if and only if there is an agent i that is present in all winning coalitions, i.e., $i \in \cap_{\nu(S)=1} S$. Moreover, if the core of G is non-empty, then the nucleolus of G is given by $x_i = 1/k$ if $i \in \cap_{\nu(S)=1} S$ and $x_i = 0$ otherwise, where $k = |\{i : i \in \cap_{\nu(S)=1} S\}|$.*

Proof. The first part is straightforward, so we prove that if the core of G is non-empty, then the imputation $\mathbf{x}$ described in the statement of the theorem is indeed the nucleolus of G. Let $M = \{i : i \in \cap_{\nu(S)=1} S\}$. Any imputation $(p_1, \dots, p_n)$ that has $p_i > 0$ for some $i \notin M$ is not in the core of G, as there exists a set S with $\nu(S) = 1$, $i \notin S$, for which we have $e(\mathbf{p}, S) \leq -p_i$. Hence, as the nucleolus $\mathbf{x}$ is always in the core, it satisfies $x_i = 0$ for all $i \notin M$. Now, consider a vector $\mathbf{p}$ with $p_i = 0$ for

$i \notin M$, and suppose that $p_i \neq 1/k$ for some $i \in M$. Let $j = \text{argmin}\{p_i \mid i \in M\}$. We have $p_j < 1/k$. Let $t = |\{S : \nu(S) = 1\}| + |\{S : S \subseteq I \setminus M\}|$. The excess vectors for $\mathbf{p}$ and $\mathbf{x}$ start with t zeros, followed by p_j and $1/k$, respectively. Hence, the excess vector for $\mathbf{p}$ is lexicographically smaller than the excess vector for $\mathbf{x}$. □

Remark 2. *It is easy to check if there is a agent that is present in all winning coalitions. Namely, for each agent i, we check if $w(I \setminus \{i\}) \geq T$; if this is not the case, $i \in \cap_{w(S) \geq T} S$.*

Consider the following computational problems.

Name LEASTCORE.
Instance Weighted threshold game $(I; \mathbf{w}; T)$, and rational value $\epsilon \geq 0$.
Question Is the ϵ-core of $(I; \mathbf{w}; T)$ non-empty?

The smallest ϵ for which (G, ϵ) is a "yes"-instance of LEASTCORE corresponds to the least core of G.

Name IN-LEASTCORE.
Instance Weighted threshold game $(I; \mathbf{w}; T)$, imputation $\mathbf{p}$.
Question Is $\mathbf{p}$ in the least core of $(I; \mathbf{w}; T)$?

Name CONSTRUCT-LEASTCORE.
Instance Weighted threshold game $(I; \mathbf{w}; T)$.
Output An imputation $\mathbf{p}$ in the least core of $(I; \mathbf{w}; T)$.

We now show that the problems LEASTCORE, IN-LEASTCORE, and CONSTRUCT-LEASTCORE are NP-hard. We reduce from the well-known NP-complete PARTITION problem, in which we are given positive integers $a_1, \dots, a_n$ such that $\sum_{i=1}^{n} a_i = 2K$, and asked whether there is a subset of indices J such that $\sum_{i \in J} a_i = K$ (Garey & Johnson 1979, p.223).

Given an instance $(a_1, \dots, a_n; K)$ of PARTITION, let $I = \{1, \dots, n, n+1\}$ be a set of agents. Let $G = (I; \mathbf{w}; T)$ be the weighted threshold game with $T = K$, $w_i = a_i$ for $i = 1, \dots, n$ and $w_{n+1} = K$. We will use the following lemmas.

Lemma 3. *For a "yes"-instance of* PARTITION*, the least core of G is its $2/3$-core, and any imputation $\mathbf{q} = (q_1, \dots, q_{n+1})$ in the least core satisfies $q_{n+1} = 1/3$.*

Proof. Consider the imputation $\mathbf{p}$ given by $p_i = \frac{w_i}{3K}$ for $i = 1, \dots, n+1$ (this is a valid imputation, as $\sum_{i=1}^{n+1} w_i = 3K$). For any set S with $\nu(S) = 1$ we have $\sum_{i \in S} w_i \geq K$, so $\sum_{i \in S} p_i \geq 1/3$ and $e(\mathbf{p}, S) \geq -2/3$; for any set S with $\nu(S) = 0$ we have $e(\mathbf{p}, S) \geq 0$. We conclude that the least core of G is contained in its $2/3$-core, i.e., the least core of G is its ϵ-core for some $\epsilon \leq 2/3$.

On the other hand, for a "yes"-instance of PARTITION, there are three disjoint coalitions in I that have value 1: $S_1 = J$, $S_2 = \{1, \dots, n\} \setminus J$, and $S_3 = \{n+1\}$. Any imputation $\mathbf{p}$ such that $p_{n+1} \neq 1/3$ has $p(S_i) < 1/3$ for some $i = 1, 2, 3$ and hence $e(\mathbf{p}, S_i) < -2/3$. Hence, any imputation $\mathbf{q}$ that maximizes the minimum excess satisfies $q_{n+1} = 1/3$. Consequently, the value of ϵ that corresponds to the least core satisfies $\epsilon = 2/3$, and any imputation in the least core has $q_{n+1} = 1/3$. □

Lemma 4. *For a "no"-instance of* PARTITION*, the least core of G is its ϵ-core for some $\epsilon < 2/3$ and any imputation $\mathbf{q}$ in the least core satisfies $q_{n+1} > 1/3$.*

Proof. We will start with the imputation $p_i = \frac{w_i}{3K}$ defined in the proof of the previous lemma, and modify it so as to ensure that for a new imputation $\mathbf{p}'$, the excess of each coalition is strictly greater than $-2/3$. The imputation $\mathbf{p}'$ will serve as a witness that the least core of G is its ϵ-core for $\epsilon < 2/3$. Consequently, for any imputation $\mathbf{q}$ in the least core we have $e(\mathbf{q}, S) > -2/3$ for any $S \subseteq I$. In particular, taking $S = \{n+1\}$, we obtain $q_{n+1} > 1/3$.

The imputation $\mathbf{p}'$ is defined as follows: $p'_i = p_i - \frac{1}{6nK}$ for $i = 1, \dots, n$, $p'_{n+1} = p_{n+1} + \frac{1}{6K}$. To see that $\mathbf{p}'$ is a valid imputation, note that $\sum_{i \in I} p'_i = \sum_{i \in I} p_i = 1$, and $p'_i = \frac{w_i}{3K} - \frac{1}{6nK} > 0$. Now, consider any set S such that $\nu(S) = 1$. If $S \subseteq \{1, \dots, n\}$, as our game was constructed from a "no"-instance of PARTITION, we have $\sum_{i \in S} w_i \geq K + 1$. Hence,

$$\sum_{i \in S} p'_i = \sum_{i \in S} \left(\frac{w_i}{3K} - \frac{1}{6nK} \right) = \frac{1}{3} + \frac{1}{6K}.$$

Consequently, $e(\mathbf{p}', S) > -2/3$. On the other hand, if $n+1 \in S$, we have $p'(S) > \frac{1}{3} + \frac{1}{6K}$, so again $e(\mathbf{p}', S) > -2/3$. □

Theorem 5. *The problems* LEASTCORE*,* IN-LEASTCORE*, and* CONSTRUCT-LEASTCORE *are NP-hard.*

Proof. By combining Lemmas 3 and 4, we conclude that if we can decide whether the $2/3 - 1/(6K)$-core of $G = (I; \mathbf{w}; T)$ is nonempty then we can correctly solve PARTITION. Also, if we can construct a solution in the least core, we can solve PARTITION by looking at its last component q_{n+1}. Finally, the imputation $\mathbf{p}$, where $p_i = \frac{w_i}{3K}$, is in the least core if and only if the game G was constructed from a "yes"-instance of PARTITION. Hence, correctly deciding whether $\mathbf{p}$ is in the least core allows us to solve PARTITION as well. □

Pseudopolynomial time algorithm for the least core

The following theorem gives a pseudopolynomial time algorithm for the problems CONSTRUCT-LEASTCORE, IN-LEASTCORE and LEASTCORE. This means that all three problems can be solved in polynomial time if the weights are bounded (at most polynomially large in n), or (equivalently) if they are represented in unary notation.

Theorem 6. *If all weights are represented in unary, the problems* CONSTRUCT-LEASTCORE*,* IN-LEASTCORE *and* LEASTCORE *are in P.*

Proof. Consider the following linear program:

$$\begin{aligned} \max \quad C; \\ p_1 + \dots + p_n = 1 \\ p_i \geq 0 \text{ for all } i = 1, \dots, n \\ \sum_{i \in J} p_i \geq C \text{ for all } J \subseteq I \text{ such that } \sum_{i \in J} w_i \geq T \end{aligned} \quad (1)$$

This linear program attempts to maximize the minimum excess by computing the greatest lower bound C on the payment to each winning coalition (i.e., a coalition whose total weight is at least T). Any solution to this linear program is a vector of the form $(p_1, \dots, p_n, C)$; clearly, the imputation $(p_1, \dots, p_n)$ is in the least core, which coincides with the $(1 - C)$-core.

Unfortunately, the size of this linear program may be exponential in n, as there is a constraint for each winning coalition. Nevertheless, we will now show how to solve it in time polynomial in n and $\sum_{i \in I} w_i$, by constructing a *separation oracle* for it. A separation oracle for a linear program is an algorithm that, given an alleged feasible solution, checks whether it is indeed feasible, and if not, outputs a violated constraint (Schrijver 2003). It is known that a linear program can be solved in polynomial time as long as it has a polynomial-time separation oracle. In our case, this means that we need an algorithm that given a pair $((p_1, \dots, p_n), C)$, checks if there is a winning coalition J such that $\sum_{i \in J} p_i < C$.

To construct the separation oracle, we will use dynamic programming to determine $P_0 = \min p(J)$ over all winning coalitions J. If $P_0 < C$, then the constraint that corresponds to $\mathrm{argmin}_{w(J) \geq T}\, p(J)$ is violated. Let $W = \sum_{i \in I} w_i$. For $k = 1, \dots, n$ and $w = 1, \dots, W$, let $x_{k,w} = \min\{p(J) \mid J \subseteq \{1, \dots, k\}, w(J) = w\}$. Clearly, we have $P_0 = \min_{w=T,\dots,W} x_{n,w}$. It remains to show how to compute $x_{k,w}$. For $k = 1$, we have $x_{1,w} = p_1$ if $w = w_1$ and $x_{1,w} = +\infty$ otherwise. Now, suppose we have computed $x_{k,w}$ for all $w = 1, \dots, W$. Then we can compute $x_{k+1,w}$ as $\min\{x_{k,w}, p_{k+1} + x_{k,w-w_k}\}$. The running time of this algorithm is polynomial in n and W, i.e., in the size of the input.

Now, consider the application of the linear program for a weighted threshold game $G = (I; \mathbf{w}; T)$. The constructed imputation $\mathbf{p}$ is a solution for CONSTRUCT-LEASTCORE with instance G. Also, the solution to LEASTCORE with instance G, ϵ should be "yes" iff $\epsilon = C - 1$. The solution to IN-LEASTCORE with instance G, $\mathbf{p}'$ should be "yes" if and only if every winning coalition $S \subseteq I$ has $\mathbf{p}'(S) \geq C$. This can be checked in polynomial time using the separation oracle from the proof of Theorem 6. □

Approximation scheme for the least core

In this section, we show that the pseudopolynomial algorithm of the previous section can be converted into an approximation scheme. More precisely, we construct an algorithm that, given a game $G = (I; \mathbf{w}; T)$ and a $\delta > 0$, outputs ϵ' such that if the least core of G is equal to its ϵ-core then $\epsilon \leq \epsilon' \leq \epsilon + 2\delta$. The running time of our algorithm is polynomial in the size of the input as well as $1/\delta$, i.e., it is a fully polynomial additive approximation scheme. Subsequently, we show that it can be modified into a fully polynomial multiplicative approximation scheme (FPTAS), i.e., an algorithm that outputs ϵ' satisfying $\epsilon \leq \epsilon' \leq (1 + \delta)\epsilon$.

Consider the linear program (1) in which C is some fixed integer multiple of δ and the goal is to find a feasible solution for this value of C or report than none exists. It is known that problems of this type can be solved in polynomial time as long as they have a polynomial-time separation oracle. We will describe a subroutine $\mathcal{A}$ that, given C, either outputs a feasible solution $\mathbf{p}$ for $C - \delta$ or correctly solves the problem for C. Our algorithm runs $\mathcal{A}$ for $C = 0, \delta, 2\delta, \dots, 1$ and outputs $\epsilon' = 1 - C'$, where C' is the maximum value of C for which $\mathcal{A}$ finds a feasible solution.

Clearly, we have $\epsilon \leq 1 - C'$. Now, let $C^* = 1 - \epsilon$ be the optimal solution to the original linear program and let $k^* = \max\{k \mid k\delta \leq C^*\}$. As $k^*\delta \leq C^*$, there is a feasible solution for $k^*\delta$. When $\mathcal{A}$ is given $k^*\delta$, it either solves the linear program correctly, i.e., finds a feasible solution for $k^*\delta$, or finds a feasible solution for $k^*\delta - \delta$. In any case, we have $C' \geq (k^* - 1)\delta \geq C^* - 2\delta$, i.e., $1 - C' \leq \epsilon + 2\delta$.

It remains to describe the subroutine $\mathcal{A}$. Given a $C = k\delta$, it attempts to solve the linear program using the ellipsoid method. However, whenever the ellipsoid method calls the separation oracle for some payoff vector $\mathbf{p}$, we simulate it as follows. We set $\delta' = \delta/n$ and round down $\mathbf{p}$ to the nearest multiple of δ', i.e., set $p'_i = \max\{j\delta' \mid j\delta' \leq p_i\}$. We have $0 \leq p_i - p'_i \leq \delta'$. Let $x_{i,j} = \max\{w(J) \mid J \subseteq \{1, \dots, j\}, \mathbf{p}'(J) = i\delta'\}$. The values $x_{i,j}$ are easy to compute by dynamic programming. Consider $U = \max\{x_{i,n} \mid i = 1, \dots, (k-1)n - 1\}$. This is the maximum weight of a coalition whose total payoff under $\mathbf{p}'$ is at most $c - \delta - \delta'$. Since payments increment by δ' this is the maximum weight of a coalition whose total payoff is less than $C - \delta$. If $U < T$, the payoff to each winning coalition under $\mathbf{p}'$ is at least $C - \delta$; as $p_i > p'_i$, the same is true for $\mathbf{p}$. Hence, $\mathbf{p}$ is a feasible solution for $C - \delta$, so $\mathcal{A}$ outputs $\mathbf{p}$ and stops.

If $U \geq T$, there exists a winning coalition J such that $\mathbf{p}'(J) < C - \delta$ and hence $\mathbf{p}(J) < C$; moreover, this J can be found using standard dynamic programming techniques. This means that that we have found a violated constraint, i.e., successfully simulated the separation oracle and can continue with the ellipsoid method.

Remark 7. *It is easy to verify that if the least core of $G = (I; \mathbf{w}; T)$ is its ϵ-core, then we have $\epsilon \geq 1/|I|$. This means that the algorithm described above can be converted into an FPTAS; we omit the details.*

The Nucleolus

Consider the following computational problems:

Name NUCLEOLUS.
Instance Weighted threshold game $(I; \mathbf{w}; T)$, agent $i \in I$.
Output The nucleolus payoff of agent i in $(I; \mathbf{w}; T)$.

Name ISZERO-NUCLEOLUS.
Instance Weighted threshold game $(I; \mathbf{w}; T)$, agent $i \in I$.
Question Is the nucleolus payoff of agent i in $(I; \mathbf{w}; T)$ equal to 0?

We will show that ISZERO-NUCLEOLUS is NP-hard. Clearly, this implies that NUCLEOLUS is NP-hard as well. We start with the following lemma.

Lemma 8. *For weighted threshold games, if the Shapley value of a agent is 0, his nucleolus payoff is 0 as well, i.e., $\phi(i) = 0$ implies $x_i = 0$.*

Proof. For the Shapley value of a agent i to be 0, it has to be the case that $\nu(S) = \nu(S \cup \{i\})$ for all $S \subseteq I$. Now, suppose that $\phi(i) = 0$, but $x_i \neq 0$, and consider the excess vector for $\mathbf{x}$. Let $e(\mathbf{x}, S)$ be the first element of this vector; clearly, $\nu(S) = 1$. It is easy to see that $i \notin S$: otherwise, we would have $\nu(S \setminus \{i\}) = 1$ and moreover, $x(S \setminus \{i\}) = x(S) - x_i < x(S)$. Now, consider an imputation $\mathbf{q}$ given by $q_i = \frac{x_i}{2}$, $q_j = x_j + \frac{x_i}{2(n-1)}$ for $j \neq i$. For any non-empty coalition T such that $i \notin T$ we have $q(T) > x(T)$. Moreover, as $q_i \neq 0$, using the same argument as for $\mathbf{x}$, we conclude that the first element of the excess vector $e(\mathbf{q}, T)$ satisfies $i \notin T$. Hence,

$$e(\mathbf{q}, T) = q(T) - \nu(T) > x(T) - \nu(T) = e(\mathbf{x}, T) \geq e(\mathbf{x}, S),$$

a contradiction with the definition of the nucleolus. □

Remark 9. *The converse of Lemma 8 is not true. Consider the coalitional game with $I = \{1, 2, 3\}$, $\mathbf{w} = \{\frac{1}{2}, \frac{1}{4}, \frac{1}{4}\}$ and $T = \frac{3}{4}$. Winning coalitions are those that contain agent 1 and at least one of agents 2 and 3. The Shapley value $\phi(3)$ is positive because there is a positive coalition from the coalition $S = \{1, 3\}$ (see the definition of ϕ). However, by Theorem 1, the nucleolus payoff $x_3 = 0$.*

Theorem 10. *The problem* ISZERO-NUCLEOLUS *is NP-hard.*

Proof. As in the proof of Theorem 5, we construct a weighted threshold game based on an instance of PARTITION. Given an instance $A = (a_1, \ldots, a_n; K)$ of PARTITION, let $G = (I; \mathbf{w}; T)$ be the weighted threshold game with $I = \{1, \ldots, n, n+1\}$, $T = K + 1$, $w_{n+1} = 1$, and $w_i = a_i$ for $i = 1, \ldots, n$. We will show that $x_{n+1} \neq 0$ if and only if A is a "yes"-instance of PARTITION.

Suppose first that A is a "no"-instance of PARTITION. Consider any winning coalition $S \subseteq I$ such that $n + 1 \in S$. We have $w(S) \geq K + 1$. Moreover, if $w(S) = K + 1$, then $w(S \setminus \{n+1\}) = K$, implying that there is a partition. Hence, $w(S) > K + 1$, or, equivalently, $\nu(S \setminus \{n+1\}) = 1$. We conclude that the Shapley value of the $(n+1)$st agent is 0. By Lemma 8, this implies $x_{n+1} = 0$.

Now, suppose that A is a "yes"-instance of PARTITION. Let $I' = I - \{n+1\}$ and let J be a partition of I' satisfying $w(J) = w(I' \setminus J) = K$. Consider an imputation $\mathbf{p}$ with $p_{n+1} = 0$. The sets $S_1 = J \cup \{n+1\}$ and $S_2 = (I' \setminus J) \cup \{n+1\}$ satisfy $\nu(S_1) = \nu(S_2) = 1$. As $p_{n+1} = 0$, we have $p(S_1) + p(S_2) = p(J) + p(I' \setminus J) = 1$, so $\min\{e(\mathbf{p}, S_1), (\mathbf{p}, S_2)\} \leq -1/2$. That is, for any imputation with $p_{n+1} = 0$ the minimum excess is at most $-1/2$. On the other hand, under the imputation $q_i = \frac{w_i}{2K+1}$ the payoff of each winning coalition is at least $\frac{K+1}{2K+1} > 1/2$, i.e., for this imputation the minimum excess is strictly greater than $-1/2$. As we have $\min_{S \subseteq I} e(\mathbf{x}, S) \geq \min_{S \subseteq I} e(\mathbf{q}, S)$, we conclude that $x_{n+1} \neq 0$. □

Remark 11. *Theorem 10 implies that the problem* NUCLEOLUS *cannot be approximated within any constant factor unless P=NP. More formally, it is not in the complexity class APX (Ausiello* et al. *1999, p.91) unless P=NP. The same holds for the problem of computing the Shapley value.*

Remark 12. *We can use the construction in the proof of Theorem 5 to show that* NUCLEOLUS *is NP-hard; however, it does not imply the NP-hardness of* ISZERO-NUCLEOLUS*. Conversely, the proof of Theorem 10 does not immediately imply that the least core-related problems are NP-hard. Therefore, to prove that all of our problems are NP-hard, we need both constructions.*

Remark 13. *While we have proved that the problems considered in this subsection are NP-hard, it is not clear that they are in NP. Consider, for example,* ISZERO-NUCLEOLUS*. To verify that the nucleolus payoff of an agent i is 0, we would have to prove that there is an imputation $\mathbf{x}$ (the nucleolus) with $x_i = 0$, and that any imputation $\mathbf{p}$ with $p_i > 0$ produces an excess vector that is lexicographically smaller than that of $\mathbf{x}$. The latter condition involves exponentially-long vectors.*

Approximating the Nucleolus

Without loss of generality, we can assume that the sum of the weights in a weighted threshold game is 1. We will refer to such a game as a *normalised* weighted threshold game. Note that any weighted threshold game is equivalent to some normalised game.

For many normalised weighted threshold games considered in the literature, the vector $\mathbf{w}$ coincides with the nucleolus. For example, consider the set $\mathcal{C}$ of *constant-sum* games. A normalised weighted threshold game $G = (I; \mathbf{w}; T)$ is in $\mathcal{C}$ if, for any $S \subseteq I$, $\nu(S) + \nu(I \setminus S) = 1$. (Peleg 1968) shows the following. Suppose $G = (I; \mathbf{w}; T) \in \mathcal{C}$. Let x be the nucleolus for G and let $G' = (I; \mathbf{x}; T)$. Then the nucleolus of G' is also equal to $\mathbf{x}$. (Wolsey 1976) shows a simlar result for the set $\mathcal{C}'$ of *symmetric* games. A normalised weighted threshold game $G = (I; \mathbf{w}; T)$ is in $\mathcal{C}'$ if $T = 1/2$ and there is a coalition S with $\nu(S) = \nu(I \setminus S)$.

It is not true in general that the vector $\mathbf{w}$ coincides with the nucleolus. It is also not true that w_i is a good approximation to the nucleolus payoff x_i. For example, in the game considered in Remark 9 the nucleolus payment x_3 is 0 but $w_3 = \frac{1}{4}$ (so these are not related by a constant factor). The nucleolus payment x_i can also exceed w_i by an arbitrary factor. For example, take an arbitrarily small $\delta > 0$. Consider the game with $I = \{1, 2\}$, $\mathbf{w} = \{1 - \delta, \delta\}$, and $T = 1 - \delta/2$. By Theorem 1, $x = (0.5, 0.5)$ so $x_2 = 0.5$. In any case, it is clear from Corollary 11 that w_i cannot be a constant-factor approximation to the nucleolus payment x_i of an individual agent i, since that would imply P=NP.

In Theorem 20 we show that, for an appropriate sense of approximation based on *coalitions* rather than on *individual agents*, the vector $\mathbf{w}$ provides a good *approximation* to the nucleolus. Our result applies to a large class of weighted threshold games. We start with a simple lower bound on nucleolus payments.

Lemma 14. *Let $G = (I; \mathbf{w}; T)$ be a normalised weighted threshold game. If $w(S) \geq T$ then $x(S) \geq T$.*

Proof. A winning coalition S has $e(\mathbf{w}, S) = \mathbf{w}(S) - 1 \geq T - 1$. The nucleolus maximizes the minimum payoff to a winning coalition, so $e(\mathbf{x}, S) \geq T - 1$ and $\mathbf{x}(s) \geq T$. □

A *minimal* winning coalition is a coalition S with $\mathbf{w}(S) \geq T$ for which every proper subset $S' \subset S$ has $\mathbf{w}(S') < T$. We will now use Lemma 14 to show that the weight of any minimal winning coalition is at most twice its nucleolus payoff.

Lemma 15. *Let $G = (I; \mathbf{w}; T)$ be a normalised weighted threshold game. Suppose that every agent $i \in I$ has $w_i \leq T$. Let $S \subseteq I$ be a minimal winning coalition in G. Then $w(S) < 2x(S)$.*

Proof. Let i be an agent in S. Since S is minimal, $w(S \setminus \{i\}) < T$. So $w(S) < T + w_i < 2T$. The result now follows from Lemma 14. □

We do not know whether there is a value α such every minimal winning coalition S of a normalised weighted threshold game satisfies $x(S) \leq \alpha w(S)$. However, it is easy to see that this is true with $\alpha = 2$ if $T \geq 1/2$ since $x(S) \leq 1$ and, for a winning coalition S, $w(S) \geq T \geq 1/2$. So we get the following observation.

Observation 16. *Let $G = (I; \mathbf{w}; T)$ be a normalised weighted threshold game with $T \geq 1/2$. Let $S \subseteq I$ be a winning coalition in G. Then $x(S) \leq 2w(S)$.*

If T is less than $1/2$ but is relatively large compared to the individual weights, the vector $\mathbf{w}$ is still a good approximation to the nucleolus.

Lemma 17. *Consider a normalised weighted threshold game $G = (I; \mathbf{w}; T)$ that satisfies $w_i \leq \epsilon T$, $T \geq \frac{\epsilon}{1+\epsilon}$ for some $\epsilon \leq 1$. For any such game, any minimal winning coalition $S \subseteq I$ satisfies $x(S) \leq 3w(S)$*

Proof. For any minimal winning coalition S, we have $w(S \setminus \{i\}) < T$ for any $i \in S$, so $w(S) < T + w_i < T(1+\epsilon)$. Now, fix a minimal winning coalition S_0. We have $w(S_0) \geq T$, $w(I \setminus S_0) > 1 - T(1+\epsilon)$. We can construct a collection of $t = \lfloor \frac{1-T(1+\epsilon)}{T(1+\epsilon)} \rfloor \geq \frac{1}{T(1+\epsilon)} - 2$ disjoint minimal winning coalitions in $I \setminus S$. (For example, we can construct these coalitions consecutively by adding agents to a current coalition one by one until the weight of the coalition under construction becomes at least T.) Let these coalitions be $S_1, \ldots, S_t$. Lemma 14 implies $x(S_i) \geq T$ for $i = 1, \ldots, t$. Hence, $x(S_0) \leq 1 - tT \leq 2T - \frac{1}{1+\epsilon} + 1 \leq 2T + \frac{\epsilon}{1+\epsilon} = 3T \leq 3w(S_0)$. □

Remark 18. *Let $G = (I; \mathbf{w}; T)$ be a normalised weighted threshold game which satisfies $w_i \leq T^2$ for every agent $i \in I$. Then Lemma 17 applies with $\epsilon = T$.*

Remark 19. *By setting $\epsilon = 1$ in Lemma 17, we can obtain that $x(S) \leq 3w(S)$ for $T \geq 1/2$ with the additional restriction that $w_i \leq T$ for all w_i; considering the case $T \geq 1/2$ separately using Observation 16 gives us a stronger result.*

Lemma 15, Observation 16 and Lemma 17 give us the following theorem. The theorem shows that, for a wide class of normalised weighted threshold games, the weight vector $\mathbf{w}$ approximates the nucleolus $\mathbf{x}$ in the sense that the payoff to a minimal winning coalition only differs by at most a factor of 3 in these two imputations.

Theorem 20. *Let $G = (I; \mathbf{w}; T)$ be a normalised weighted threshold game. Suppose that every agent $i \in I$ has $w_i \leq T$. If $T \geq 1/2$ then any minimal winning coalition S satisfies $w(S)/2 \leq x(S) \leq 2w(S)$. If there is an $\epsilon \in (0, 1]$ such that $T \geq \frac{\epsilon}{1+\epsilon}$ and every agent satisfies $w_i \leq \epsilon T$ then any minimal winning coalition S satisfies $w(S)/2 \leq x(S) \leq 3w(S)$.*

References

Ausiello, G.; Crescenzi, P.; Gambosi, G.; Kann, V.; Marchetti-Spaccamela, A.; and Protasi, M. 1999. *Complexity and Approximation*. Springer-Verlag: Berlin, Germany.

Bilbao, J. M.; Fernández, J. R.; Jiminéz, N.; and López, J. J. 2002. Voting power in the European Union enlargement. *European Journal of Operational Research* 143:181–196.

Conitzer, V., and Sandholm, T. 2006. Complexity of constructing solutions in the core based on synergies among coalitions. *Artificial Intelligence* 170:607–619.

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

Garey, M. R., and Johnson, D. S. 1979. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. New York, NY, USA: W. H. Freeman & Co.

Ieong, S., and Shoham, Y. 2005. Marginal contribution nets: A compact representation scheme for coalitional games. In *Proceedings of the Sixth ACM Conference on Electronic Commerce (EC'05)*.

Matsui, T., and Matsui, Y. 2000. A survey of algorithms for calculating power indices of weighted majority games. *J. Oper. Res. Soc. Japan* 43(1):71–86.

Matsui, Y., and Matsui, T. 2001. NP-completeness for calculating power indices of weighted majority games. *Theoret. Comput. Sci.* 263(1-2):305–310.

Osborne, M. J., and Rubinstein, A. 1994. *A Course in Game Theory*. The MIT Press: Cambridge, MA.

Papadimitriou, C. H. 1994. *Computational Complexity*. Addison-Wesley: Reading, MA.

Peleg, B. 1968. On weights of constant-sum majority games. *SIAM J. Appl. Math.* 16:527–532.

Prasad, K., and Kelly, J. S. 1990. NP-completeness of some problems concerning voting games. *Internat. J. Game Theory* 19(1):1–9.

Schrijver, A. 2003. *Combinatorial Optimization: Polyhedra and Efficiency*, volume 24 of *Algorithms and Combinatorics*. Springer.

Wolsey, L. A. 1976. The nucleolus and kernel for simple games or special valid inequalities for $0-1$ linear integer programs. *Internat. J. Game Theory* 5(4):227–238.