Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence

Computing the Nucleolus of Matching, Cover and Clique Games

Ning Chen
Division of Mathematical Sciences
Nanyang Technological University
Singapore

Pinyan Lu
Microsoft Research Asia
Beijing, China

Hongyang Zhang
Department of Computer Science
Shanghai Jiao Tong University
Shanghai, China

Abstract

In cooperative games, a key question is to find a division of payoffs to coalition members in a fair manner. Nucleolus is one of such solution concepts that provides a stable solution for the grand coalition. We study the computation of the nucleolus of a number of cooperative games, including fractional matching games and fractional edge cover games on general weighted graphs, as well as vertex cover games and clique games on weighted bipartite graphs. Our results are on the positive side—we give efficient algorithms to compute the nucleolus, as well as the least core, of all of these games.

Introduction

A central question in cooperative game theory is how to distribute a certain amount of profit generated by a group of agents N, denoted by a function $f(N)$, to each individual. A number of solution concepts have been proposed to capture fairness of the distribution (a.k.a. imputation) among the agents. One of the most important notions is that of *core*, which requires that no group of agents can benefit by breaking away from the grand coalition. That is, for any $S \subseteq N$, $\sum_{i \in S} x_i \geq f(S)$, where x_i is the share that agent i obtains.

In many games, however, the core may be empty. Further, even if the core is nonempty, it may not be a singleton; thus, it is still unclear how the benefit of cooperation should be shared. The notion of *nucleolus* was introduced in (Schmeidler 1969) as a single point solution for cooperative games. Roughly speaking, it is the unique distribution that lexicographically maximizes the vector of non-decreasingly ordered excesses, defined as $\sum_{i \in S} x_i - f(S)$, over the set of imputations. It is well known that if the core is nonempty, it always contains the nucleolus. Nucleolus provides a grand coalition solution of a game, and has been applied in, e.g., insurance and bankruptcy policies (Lemaire 1984; Aumann and Maschler 1985).

Due to its vast applications, nucleolus has received much attention in the past a few decades. Following the definition, (Kopelowitz 1967), as well as (Maschler, Peleg, and Shapley 1979), proposed an approach based on solving a sequence of linear programs to find the nucleolus. While there are at most N such linear programs to solve, constructing one may take exponential steps due to the number of constraints corresponding to all possible coalitions. Therefore, it is in general far from clear how to apply this method, and computational complexity has been taken into account for finding the nucleolus for various cooperative games; see, e.g., (Deng and Papadimitriou 1994).

The first efficient (i.e., polynomial time) algorithm computing the nucleolus was proposed in (Megiddo 1978) for minimum spanning tree games when the underlying graph is a tree. Later on, a number of efficient algorithms were developed for, e.g., matching games (Solymosi and Raghavan 1994; Kern and Paulusma 2003; Biro, Kern, and Paulusma 2012), standard tree games (Granot et al. 1996), flow games (Deng, Fang, and Sun 2009), voting games (Elkind and Pasechnik 2009), bankruptcy games (Aumann and Maschler 1985), airport profit games (Branzei et al. 2006), and spanning connectivity games (Aziz et al. 2009). On the other hand, NP-hardness results for computing the nucleolus were shown for, e.g., minimum spanning tree games (Faigle, Kern, and Kuipers 1998), threshold games (Elkind et al. 2007), flow and linear production games (Deng, Fang, and Sun 2009).

We follow the stream and study computing the nucleolus of some combinatorial games. We first study *fractional matching games*, where given a general weighted graph with weights on edges, agents correspond to vertices and the valuation of a coalition S is the value of a maximum weighted fractional matching on S. Fractional matching games generalize the classic assignment games (Shapley and Shubik 1971) and takes into account the fact that agents usually split their collaborations fractionally with each other. For instance, in our daily life, one usually splits his or her available time to several tasks that involve other agents; fractional matching games consider how to allocate the time from a game theoretical viewpoint. Fractional matchings games are a special case of the more general linear production games (Owen 1975) for which the core is always nonempty. We show that the nucleolus, as well as the least core, can be computed in polynomial time for fractional matching games.

Note that for bipartite graphs, as an optimal fractional solution is integral, our result immediately implies an efficient algorithm computing the nucleolus for bipartite integral matching games. Such a result for bipartite graphs was shown previously by (Solymosi and Raghavan 1994); but our algorithm is much simpler and applies to a more general setting. (Kern and Paulusma 2003) studied matching games for general graphs as well, but they considered unweighted graphs and integral matchings, which are different from ours. In a recent work, (Biro, Kern, and Paulusma 2012) gave an $O(n^4)$ algorithm that computes the nucleolus of integral matching games on general weighted graphs given that the core is nonempty. Note that for general weighted graphs, computing the nucleolus of integral matching games has been a long standing open problem.

We next consider *vertex cover games*, introduced in (Deng, Ibaraki, and Nagamochi 1999), where we are given a graph with weights on vertices and agents corresponding to edges. The valuation of a coalition S is the value of a minimum vertex cover for S. (For such games, valuations are interpreted as costs and agents would like to minimize their assigned share.) Vertex cover games may find applications in, e.g., deciding the locations of facilities. We show that for bipartite graphs, the least core and the nucleolus of vertex cover games can be computed in polynomial time.

Both of our algorithms, at a high level view, are based on the linear programming approach of computing nucleolus and careful exploitations into the combinatorial structures of the problems.

Preliminaries

A cooperative game is defined by a set of agents N and a characteristic function $f : 2^N \to \mathbb{R}$, associating a value $f(S)$ to every subset $S \subseteq N$, where $f(\emptyset) = 0$. A central notion in cooperative games is that of *core*, defined as below:

$$core(N, v) = \left\{ x \in \mathbb{R}^N \mid \begin{array}{l} x(N) = f(N), \\ x(S) \geq f(S),\ \forall\, S \subseteq N \end{array} \right\}$$

where $x(S) = \sum_{i \in S} x_i$ is the allocation of coalition S. The intuition of the definition is that all agents in N share the total valuation $f(N)$ while no subset of agents are willing to deviate. Throughout the paper, we will use N to denote the set of agents and i to denote an agent.

The *excess* of a coalition $S \subseteq N$ with respect to an allocation $x \in \mathbb{R}^N$ is defined as

$$e(S, x) \triangleq x(S) - f(S).$$

The *excess vector* $\theta(x)$ is defined as the $2^N - 2$ dimensional vector whose components are the excesses for the coalitions $S \subseteq N$, $S \neq \emptyset, N$, and are arranged in a non-decreasing order. The *nucleolus* is then the allocation that lexicographically maximizes the excess vector $\theta(x)$. Note that the nucleolus is unique (Schmeidler 1969).

The nucleolus can be computed by the following standard sequence of linear programs (Kopelowitz 1967; Maschler, Peleg, and Shapley 1979).

$$\begin{array}{lll} (P_1) & \max & \epsilon \\ & \text{s.t.} & x(S) \geq f(S) + \epsilon,\ \forall\, S \subseteq N, S \neq \emptyset, N \\ & & x(N) = f(N) \end{array}$$

Let ϵ_1 be the optimum value of P_1. Let $P_1(\epsilon)$ denote the set of all $x \in \mathbb{R}^N$ such that x and ϵ satisfy the constraints of P_1.[1] Hence, core$(N, v) = P_1(0)$, i.e., the core is nonempty if and only if $\epsilon_1 \geq 0$. The *least core* is defined to be the set $P_1(\epsilon_1)$, i.e., those give the optimum value for P_1.

For a given polytope $Z \subseteq \mathbb{R}^N$, let

$$fix(Z) \triangleq \left\{ S \subseteq N \mid x(S) = y(S),\ \forall\, x, y \in Z \right\}$$

denote the set of coalitions fixed by Z. Verifying if a coalition S is fixed in a bounded polytope corresponding to a linear program P can be determined by the following two linear programs.

$$\begin{array}{lll} \max & x(S) & (P_{\max}) \\ \text{s.t.} & \multicolumn{2}{l}{\text{constraints of } P} \end{array} \qquad \begin{array}{lll} \min & x(S) & (P_{\min}) \\ \text{s.t.} & \multicolumn{2}{l}{\text{constraints of } P} \end{array}$$

Note that both programs have nonempty optimal solutions since the polytope defined by P is bounded and nonempty. A coalition S is fixed if and only if the optimum values of $P_{\max}$ and $P_{\min}$ coincide.

In general, given P_{k-1} and $fix(P_{k-1}(\epsilon_{k-1}))$, we solve the following program.

$$\begin{array}{lll} (P_k) & \max & \epsilon \\ & \text{s.t.} & x(S) \geq f(S) + \epsilon,\ \forall\, S \notin fix(P_{k-1}(\epsilon_{k-1})) \\ & & x \in P_{k-1}(\epsilon_{k-1}) \end{array}$$

Denote by ϵ_k the optimum value of P_k. The process continues iteratively until a unique solution has been identified, which corresponds to the nucleolus of the game. In addition, in the iterative process, it is easy to see that $\epsilon_k \geq \epsilon_{k-1}$ and $P_k(\epsilon_k) \subseteq P_{k-1}(\epsilon_{k-1})$, for any $k > 1$.

Note that the nucleolus is an allocation and has $|N|$ variables; further, every iteration decreases the dimension of the feasible region by at least 1. Hence, in total there are at most $|N|$ linear programs to compute. However, in each iteration k we need to determine the inequality constraints corresponding to the coalitions that are not in $fix(P_{k-1}(\epsilon_{k-1}))$, which may be intractable in polynomial time. Therefore, our interest is to design efficient algorithms to find the nucleolus.

Fractional Matching Games

In this section we consider fractional matching games: Given a graph $G = (V, E)$, where the vertex set V corresponds to the set of agents N, i.e., $N = V$, there is a nonnegative weight w_e for every $e \in E$. We use $V(\cdot)$ and $E(\cdot)$ to denote the set of agents/vertices and edges, respectively, restricted on the given domain. For any coalition $S \subseteq N$, its value $f(S)$ is the value of a maximum weighted fractional matching in the subgraph induced by S, i.e.,

$$\begin{array}{lll} f(S) \triangleq & \max & \displaystyle\sum_{e \in E(S)} w_e \cdot y_e \\ & \text{s.t.} & \displaystyle\sum_{e \in E(S):\, i \in e} y_e \leq 1,\ \forall\, i \in V(S) \\ & & y_e \geq 0,\ \forall\, e \in E(S) \end{array}$$

For example, consider a triangle (V, E), where $V = \{1, 2, 3\}$, $E = \{(1, 2), (1, 3), (2, 3)\}$, and $w(1, 2) = 3$, $w(2, 3) = 2$, and $w(1, 3) = 2$. If each player uses half of his effort to cooperate with each of his neighbors, the outcome

[1]In general, $P(\epsilon)$ denotes the set of feasible solutions for the given linear program P and target value ϵ.

is 3.5, equal to the value of the maximum factional matching of the graph. The allocation $(1.5, 1.5, 0.5)$ is in $core(N, v)$. Note that if only integral matching is allowed, the problem is a classic matching game and the core may be empty.

By the formulation of the $f(S)$ defined above, it is easy to see that fractional matching game is a subclass of the more general linear production game (where the value of a coalition is obtained by solving a linear programming), for which the core is guaranteed to be nonempty (Owen 1975). This implies the following claim immediately.

Theorem 1. *The core of fractional matching games is nonempty.*

Note that for general linear production games, computing the nucleolus is NP-hard, even for special cases like flow games (Deng, Fang, and Sun 2009). For fractional matching games, however, we will show that computing the nucleolus can be solved in polynomial time.

We first need the following characterization for an optimal maximum fractional matching solution (details can be found in, e.g., (Schrijver 2003)).

Theorem 2. *For any given weighted graph G, there exists a maximum fractional matching y satisfying the following.*

- *Let $E' = \{e \in E : y_e > 0\}$. Let G' be the induced subgraph of G from edge set E'. Then the connected components of G' are either odd cycles or single edges.*
- *For any edge $e \in E'$, if e is a single edge, $y_e = 1$; if e is in an odd cycle, $y_e = \frac{1}{2}$.*

Computing the Least Core

As discussed above, the least core is captured by linear program P_1, which contains an exponential number of constraints. Here, we consider the following program (where for any edge $e = (i, j)$, we write $x(e) = x_i + x_j$):

$$
\begin{array}{lll}
(P_1^+) & \max & \epsilon \\
& \text{s.t.} & x(e) \geq w_e + \epsilon,\ \forall\ e \in E \\
& & x_i \geq \epsilon, \forall\ i \in N \\
& & x(N) = f(N)
\end{array}
$$

Let the optimal solutions of P_1^+ be ϵ_1^+. We have the following claim (recall that ϵ_1 is the optimum value of P_1).

Lemma 1. *$\epsilon_1 = \epsilon_1^+$ and the least core $P_1(\epsilon_1) = P_1^+(\epsilon_1^+)$. Thus, the least core can be computed in polynomial time.*

Proof. Note that any feasible solution of P_1 is also a feasible solution of P_1^+; therefore, $\epsilon_1^+ \geq \epsilon_1$. By Theorem 1, $\epsilon_1 \geq 0$; hence, ϵ_1^+ is non-negative. Next we will prove that for any coalition S, the excess given by any $x \in P_1^+(\epsilon_1^+)$ is at least ϵ_1^+, i.e., $x(S) - f(S) \geq \epsilon_1^+$. This immediately implies that $\epsilon_1^+ \leq \epsilon_1$, since ϵ_1 is the maximum possible least excess among all allocations.

For any $S \subseteq N$, by Theorem 2, there is a maximum fractional matching on S that can be decomposed to odd cycles and single edges; let $\mathcal{T}_1$ be the collection of odd cycles, $\mathcal{T}_2$ be the set of single edges, and $\mathcal{T}_3$ be the set of isolated (i.e., unmatched) vertices. Hence, we have

$$
\begin{aligned}
& x(S) - f(S) \\
= & \sum_{C \in \mathcal{T}_1} \big(x(V(C)) - f(V(C))\big) + \sum_{e \in \mathcal{T}_2} (x(e) - w_e) + \sum_{i \in \mathcal{T}_3} x_i \\
= & \sum_{C \in \mathcal{T}_1} \sum_{e \in E(C)} \frac{(x(e) - w_e)}{2} + \sum_{e \in \mathcal{T}_2} (x(e) - w_e) + \sum_{i \in \mathcal{T}_3} x_i
\end{aligned}
$$

For any $e \in E$ or $i \in N$, we have $x(e) - w_e \geq \epsilon_1^+$ and $x_i \geq \epsilon_1^+$ by P_1^+. Thus, the value of the formula above is at least ϵ_1^+, if $\mathcal{T}_2$ or $\mathcal{T}_3$ is nonempty, or is at least $3\epsilon_1^+/2 \geq \epsilon_1^+$, if $\mathcal{T}_1$ is nonempty (as any odd cycle contains at least three edges). This implies that $x(S) - f(S) \geq \epsilon_1^+$, and thus, $\epsilon_1^+ = \epsilon_1$.

By the definition of $P_1(\epsilon_1)$ and the above arguments, it is easy to see that $P_1(\epsilon_1) = P_1^+(\epsilon_1^+)$. □

Having determined the least core, we further identify the set of fixed constraints in $P_1^+(\epsilon_1^+) = P_1(\epsilon_1)$, which will be used in the subsequent sections. Let

$$
\begin{aligned}
E_1 & \triangleq \big\{e \in E \mid e \in fix(P_1^+(\epsilon_1))\big\} \\
N_1 & \triangleq \big\{i \in N \mid i \in fix(P_1^+(\epsilon_1))\big\}
\end{aligned}
$$

Since checking if a coalition S is fixed by P_1^+ is in polynomial and we have polynomial number of candidates to check, identifying E_1 and N_1 takes polynomial time.

Maximizing the Next Minimum Excess

Given the least core $P_1(\epsilon_1)$, to maximize the minimum excess of those coalitions that are not fixed, we next solve P_2, which, again, may have an exponential number of constraints. To get around this issue, it may be attempting to use the same idea as P_1^+, defined as below.

$$
\begin{array}{lll}
(\overline{P_2}) & \max & \epsilon \\
& \text{s.t.} & x(e) \geq w_e + \epsilon, \forall\ e \in E \setminus E_1 \\
& & x(e) = w_e + \alpha_e,\ \forall\ e \in E_1 \\
& & x_i \geq \epsilon,\ \forall\ i \in N \setminus N_1 \\
& & x_i = \alpha_i,\ \forall\ i \in N_1 \\
& & x(N) = f(N)
\end{array}
$$

where $w_e + \alpha_e$ and α_i are the fixed values for E_1 and N_1, respectively, given by $fix(P_1(\epsilon_1))$. Note that $\overline{P_2}$ was used in (Kern and Paulusma 2003) to compute the nucleolus of unweighted integral matching games with a nonempty core.

However, the following example shows that $\overline{P_2}$ fails to give the same solution as P_2 in our matching game. Let $N = \{A, B, C, D\}$, and consider the following graph.

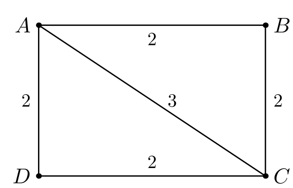

Figure 1: Graph $G = (V, E)$ with edge weights.

It can be computed that $f(N) = 4$, $\epsilon_1 = 0$, $E_1 = \{AB, BC, CD, DA\}$ with fixed value 2, and $N_1 = \emptyset$. Computing $\overline{P_2}$ gives the optimum value $\epsilon = \frac{1}{3}$ with $x_A =$

$x_C = \frac{5}{3}$ and $x_B = x_D = \frac{1}{3}$. However, consider coalition $S = \{A, B, C\}$, $S \notin fix(P_1(\epsilon_1))$ and $e(S, x) = \frac{11}{3} - \frac{7}{2} = \frac{1}{6} < \epsilon$. This implies that $\overline{P_2}$ fails to characterize $P_2(\epsilon_2)$. In the following we will show how to fix this problem.

Let G_1 be the induced subgraph with agent set $N \setminus N_1$ and edge set E_1. Let $\mathcal{C}$ be the collection of connected components of G_1. For each component $C \in \mathcal{C}$ and two vertices $i, j \in V(C)$ with $(i, j) \in E \setminus E_1$, pick *one* shortest path in C between i and j with respect to weight $x(e) - w_e$ (note that every edge in C has its $x(e)$ value fixed); denote the resulting set of shortest paths by $\mathcal{L}_1$ for all possible pairs from every component in $\mathcal{C}$.

Next we maximize the minimum excess for the coalitions that are not fixed by P_1^+. In the following program P_2^+, on the top of $\overline{P_2}$, we add a new class of constraints for the coalitions corresponding to paths in $\mathcal{L}_1$.

$$
\begin{array}{lll}
(P_2^+) & \max & \epsilon \\
& \text{s.t.} & x(e) \geq w_e + \epsilon, \forall\, e \in E \setminus E_1 \\
& & x(e) = w_e + \alpha_e,\ \forall\, e \in E_1 \\
& & x_i \geq \epsilon,\ \forall\, i \in N \setminus N_1 \\
& & x_i = \alpha_i,\ \forall\, i \in N_1 \\
& & x(V(P)) \geq f(V(P)) + \epsilon,\ \forall\, P \in \mathcal{L}_1 \\
& & x(N) = f(N)
\end{array}
$$

Lemma 2. *$\epsilon_2 = \epsilon_2^+$ and $P_2(\epsilon_2) = P_2^+(\epsilon_2^+)$, where ϵ_2 and ϵ_2^+ are the optimum value of P_2 and P_2^+, respectively.*

Proof. For any path $P \in \mathcal{L}_1$, note that while all edges in $P \subseteq E(G_1)$ are fixed, the edge between the two endpoints of P is not. Thus, $x(V(P))$ is not fixed. Hence, any feasible solution of P_2 is also feasible for P_2^+; therefore, $\epsilon_2^+ \geq \epsilon_2$.

For the other direction, it suffices to show that for any $x \in P_2^+(\epsilon_2^+)$, (x, ϵ_2^+) is also feasible for P_2. First, by the construction of P_2^+, we know that $x \in P_1^+(\epsilon_1^+)$. By Lemma 1, $P_1^+(\epsilon_1^+) = P_1(\epsilon_1)$; therefore, $x \in P_1(\epsilon_1)$. Next we show that for any $S \subseteq N$, if $S \notin fix(P_1(\epsilon_1))$, then $x(S) - f(S) \geq \epsilon_2^+$. By Theorem 2, there is a maximum fractional matching of S that can be decomposed to single edges and odd cycles, and the edges on odd cycles have optimal fractional solution 0.5; denote the set of odd cycles by $\mathcal{T}_1$, single edges by $\mathcal{T}_2$, and isolated vertices by $\mathcal{T}_3$. Then, similar to the proof of Lemma 1, we have

$$
\begin{aligned}
& x(S) - f(S) \\
= \ & \sum_{C \in \mathcal{T}_1} \sum_{e \in E(C)} \frac{x(e) - w_e}{2} + \sum_{e \in \mathcal{T}_2} (x(e) - w_e) + \sum_{i \in \mathcal{T}_3} x_i
\end{aligned}
$$

Note that $x(e) - w_e \geq 0$ for any $e \in E$, and $x_i \geq 0$ for any $i \in N$. If any $e \in \mathcal{T}_2$ is also in $E \setminus E_1$, then $x(e) - w_e \geq \epsilon_2^+$. If any $i \in \mathcal{T}_3$ is in $N \setminus N_1$, then we also have $x_i \geq \epsilon_2^+$. Otherwise, we next analyze odd cycles in $\mathcal{T}_1$. For any $C \in \mathcal{T}_1$, there are the following cases.

- If there exist at least two edges $e_1, e_2 \in E(C)$ such that both $e_1, e_2 \in E \setminus E_1$, then $x(V(C)) - f(V(C)) \geq \epsilon_2^+$.
- If there exists exactly one edge $\bar{e} = (i, j) \in E(C)$ such that $\bar{e} \in E \setminus E_1$, then consider path $E(C) \setminus \{\bar{e}\}$. Note that all edges in $E(C) \setminus \{\bar{e}\}$ are fixed and all vertices on it are not fixed (otherwise, $\bar{e}$ would be fixed as well); thus, it still exists in the graph G_1 connecting i and j. Let $P_{\bar{e}} \in \mathcal{L}_1$ be the shortest path between i and j in G_1. Then,

$$
\begin{aligned}
x(V(C)) - f(V(C)) &= \sum_{e \in E(C)} \frac{x(e) - w_e}{2} \\
&= \frac{x_{\bar{e}} - w_{\bar{e}}}{2} + \sum_{e \in E(C) \setminus \{\bar{e}\}} \frac{x(e) - w_e}{2} \\
&\geq \frac{x_{\bar{e}} - w_{\bar{e}}}{2} + \sum_{e \in P_{\bar{e}}} \frac{x(e) - w_e}{2} \\
&= x(V(P_{\bar{e}})) - \sum_{e \in P_{\bar{e}} \cup \{\bar{e}\}} \frac{w_e}{2} \\
&\geq x(V(P_{\bar{e}})) - f(V(P_{\bar{e}})) \ \geq \ \epsilon_2^+
\end{aligned}
$$

  where the last inequality follows from P_2^+.
- If there is no edge in $E(C)$ which is also in $E \setminus E_1$, then $x(V(C)) - f(V(C))$ is fixed.

Hence, either $x(S) - f(S) \geq \epsilon_2^+$, or $x(V(C)) - f(V(C))$ is fixed for all $C \in \mathcal{T}_1$. The latter implies that value $x(S) - f(S)$ is fixed, and thus, $x(S)$ is fixed in $P_1(\epsilon_1)$. Therefore, (x, ϵ_2^+) is a feasible solution of P_2 and $\epsilon_2^+ = \epsilon_2$.

To see that $P_2(\epsilon_2) = P_2^+(\epsilon_2^+)$, first $P_2(\epsilon_2) \subseteq P_2^+(\epsilon_2^+)$, since P_2^+ is a relaxation of P_2. Second, for any $x \in P_2^+(\epsilon_2^+)$, we have $x \in P_1^+(\epsilon_1^+) = P_1(\epsilon_1)$. Further, by the above arguments, $x(S) \geq f(S) + \epsilon_2$, for any $S \notin fix(P_1(\epsilon_1))$. Therefore, $x \in P_2(\epsilon_2)$. This implies that $P_2^+(\epsilon_2^+) = P_2(\epsilon_2)$. □

After solving P_2^+, again we identify the set of edges E_2 and agents N_2 (defined similar to E_1 and N_1) that are fixed in $P_2^+(\epsilon_2^+)$.

Computing the Nucleolus

Recall that the nucleolus can be computed iteratively from a series of linear programs $P_1, P_2, \ldots$, until a unique solution has been identified. We have already discussed the cases when $k = 1$ and 2. In general, for $k > 2$, similar to the definition of G_1 and $\mathcal{L}_1$, define G_{k-1} to be the subgraph induced by vertices $N \setminus N_{k-1}$ and edges E_{k-1}, and $\mathcal{L}_{k-1}$ to be the set of shortest paths in G_{k-1} connecting pairs $(i, j) \in E \setminus E_{k-1}$ with respect to weight $x(e) - w_e$, where E_{k-1} and N_{k-1} represent the set of edges and vertices in $fix(P_{k-1}(\epsilon_{k-1}))$. (Note that if multiple shortest paths exist for a given pair, only one is selected.)

Similar to P_2^+, consider the following program:

$$
\begin{array}{lll}
(P_k^+) & \max & \epsilon \\
& \text{s.t.} & x(e) \geq w_e + \epsilon, \forall\, e \in E \setminus E_{k-1} \\
& & x(e) = w_e + \alpha_e,\ \forall\, e \in E_{k-1} \\
& & x_i \geq \epsilon,\ \forall\, i \in N \setminus N_{k-1} \\
& & x_i = \alpha_i,\ \forall\, i \in N_{k-1} \\
& & x(V(P)) \geq f(V(P)) + \epsilon,\ \forall\, P \in \mathcal{L}_{k-1} \\
& & x(N) = f(N)
\end{array}
$$

where $w_e + \alpha_e$ and α_i are the fixed values for E_{k-1} and N_{k-1}, respectively, given by $fix(P_{k-1}(\epsilon_{k-1}))$.

Lemma 3. *$\epsilon_k = \epsilon_k^+$ and $P_k(\epsilon_k) = P_k^+(\epsilon_k^+)$, where ϵ_k and ϵ_k^+ are the optimum values of P_k and P_k^+, respectively.*

Proof. We prove the lemma by induction. The base case $k = 2$ has been proved in Lemma 2. For any $k > 2$, assume that $\epsilon_{k-1} = \epsilon_{k-1}^+$ and $P_{k-1}(\epsilon_{k-1}) = P_{k-1}^+(\epsilon_{k-1}^+)$; we will next prove the claim for the value k.

We first show that P_k^+ is a relaxation of P_k, that is, any feasible solution (x, ϵ) for P_k is also feasible for P_k^+. For the inequality constraints, since for any $S \in (E \setminus E_{k-1}) \cup (N \setminus N_{k-1}) \cup \mathcal{L}_{k-1}$, $x(S)$ is not fixed by P_{k-1}. Therefore $S \notin fix(P_{k-1}(\epsilon_{k-1}))$. Thus, (x, ϵ) satisfies the inequality constraints of P_k^+. For the equality constraints, they are relaxations from $x \in P_{k-1}(\epsilon_{k-1})$. By induction hypothesis, $P_{k-1}^+(\epsilon_{k-1}^+) = P_{k-1}(\epsilon_{k-1})$. Note that $E_{k-1} \cup N_{k-1} \subseteq fix(P_{k-1}^+(\epsilon_{k-1}^+))$. Therefore, (x, ϵ) satisfies the equality constraints of P_k^+. Hence, P_k^+ is a relaxation of P_k; this implies that $\epsilon_k \leq \epsilon_k^+$.

Next we prove the other direction. First, similar to the proof of Lemma 2, for any $S \subseteq N$ with $S \notin fix(P_{k-1}^+(\epsilon_{k-1}^+))$, we can show that $x(S) - f(S) \geq \epsilon_k^+$ (details omitted). Second, for any $x \in P_k^+(\epsilon_k^+)$, we will show that $x \in P_{k-1}^+(\epsilon_{k-1}^+)$. Note that in P_k^+, the constraints on edges and agents are explicitly included based on the optimal solution ϵ_{k-1}^+ of P_{k-1}^+. It suffices to consider the path constraints, say, any $P \in \mathcal{L}_{k-2}$.

- If $V(P) \notin fix(P_{k-1}(\epsilon_{k-1}))$, then
$$x(V(P)) - f(V(P)) \geq \epsilon_k^+ \geq \epsilon_k \geq \epsilon_{k-1}.$$
- If $V(P) \in fix(P_{k-1}(\epsilon_{k-1}))$, let e denote the edge between the start and end vertices of path P; then $P \cup \{e\}$ forms a cycle and $P \cup \{e\} \subseteq E_{k-1}$. Since $x(V(P))$ can be decomposed to allocations on edges and P_k^+ has kept the values of all fixed edges from $fix(P_{k-1}(\epsilon_{k-1}))$, we know that $x(V(P))$ equals the fixed value of $V(P)$ in $fix(P_{k-1}(\epsilon_{k-1}))$, which is at least $f(V(P)) + \epsilon_{k-1}$. Hence, we have $x(V(P)) \geq f(V(P)) + \epsilon_{k-1}$.

Therefore x solves P_{k-1}^+ for the optimum value ϵ_{k-1}^+. Hence, $x \in P_{k-1}^+(\epsilon_{k-1}^+)$.

Now we are ready to prove the claim for the value k. For any $x \in P_k^+(\epsilon_k^+)$, since x is also in $P_{k-1}^+(\epsilon_{k-1}^+)$ and $P_{k-1}(\epsilon_{k-1}) = P_{k-1}^+(\epsilon_{k-1}^+)$ (by induction hypothesis), we have $x \in P_{k-1}(\epsilon_{k-1})$. Further, for any $S \subseteq N$ where $S \notin fix(P_{k-1}^+(\epsilon_{k-1}^+))$, $x(S) - f(S) \geq \epsilon_k^+$; thus, (x, ϵ_k^+) solves P_k as well. Therefore, $\epsilon_k^+ \leq \epsilon_k$; this together with $\epsilon_k^+ \geq \epsilon_k$ implies that $\epsilon_k = \epsilon_k^+$.

Similar to the proof of Lemma 2, the proof for $P_k^+(\epsilon_k^+) = P_k(\epsilon_k)$ follows from repeating the above argument. This completes the proof. □

Note that P_k^+ consists of polynomial number of constraints and can be constructed easily. Further, there are at most $|N|$ iterations. We obtain the following result.

Theorem 3. *The nucleolus of a fractional matching game on weighted graphs can be computed in polynomial time.*

Vertex Cover Games

Given a graph $G = (V, E)$, for every vertex $v \in V$ there is a non-negative weight w_v. In a vertex cover game, agents $N = E$ correspond to edges. The function $f : 2^N \to \mathbb{R}$ gives the *cost* for each coalition: For any $S \subseteq N$, $f(S)$ is defined as the value of a minimum vertex cover in the subgraph induced by S. In contrast to matching games, in vertex cover games, agents want to minimize their assigned cost shares. Hence, the definitions of core, excess, and nucleolus should be changed accordingly. In addition, the sequence of linear programs P_k can be rewritten as follows.

$$\begin{aligned} (P_k') \quad & \max \quad \epsilon \\ & \text{s.t.} \quad x(S) + \epsilon \leq f(S), \forall\, S \notin fix(P_{k-1}'(\epsilon_{k-1}')) \\ & \qquad\; x \in P_{k-1}'(\epsilon_{k-1}') \\ & \qquad\; x_i \geq 0,\ \forall\, i \in N \end{aligned}$$

Let ϵ_k' denote the optimum value of P_k'.

Note that for general graphs, computing a minimum vertex cover itself is NP-hard; thus, we cannot expect to compute the core or nucleolus efficiently in the classic computational model. We will therefore focus on bipartite graphs, i.e., we assume that the underlying graph G is bipartite. For bipartite graphs, it is well known that a minimum vertex cover can be solved in polynomial time (Schrijver 2003). Further, the cost of a coalition can be computed by the following linear program, which implies that the integral optimum is equal to the fractional optimum.

$$\begin{aligned} f(S) \;=\; & \min \sum_{v \in V(S)} w_v \cdot y_v \\ & \text{s.t.} \quad y_u + y_v \geq 1,\ \forall\, e = (u, v) \in S \\ & \qquad\; y_v \geq 0,\ \forall\, v \in V(S) \end{aligned}$$

For unweighted vertex cover games on bipartite graphs, the core is always nonempty (Deng, Ibaraki, and Nagamochi 1999). The same result holds for weighted case as well, which can be shown easily by the standard primal-dual approach (details omitted).

Theorem 4. *The core of any vertex cover game on weighted bipartite graphs is nonempty.*

Computing the Least Core

For any vertex $v \in V$, let $E(v)$ denotes the set of edges (i.e., agents) adjacent to vertex v. Now we use the following linear program to compute the least core.

$$\begin{aligned} (P_1^\star) \quad & \max \quad \epsilon \\ & \text{s.t.} \quad x(E(v)) + \epsilon \leq f(E(v)),\ \forall\, v \in V \\ & \qquad\; x(N) = f(N) \\ & \qquad\; x_i \geq 0,\ \forall\, i \in N \end{aligned}$$

where $x(E(v)) = \sum_{i \in E(v)} x_i$.

Lemma 4. *$\epsilon_1' = \epsilon_1^\star$ and the least core $P_1'(\epsilon_1') = P_1^\star(\epsilon_1^\star)$, where ϵ_1' and $\epsilon_1^\star$ are the optimum of P_1' and $P_1^\star$, respectively. Thus, the least core can be computed in polynomial time.*

Proof. First, it can be seen that $\epsilon_1' \le \epsilon_1^\star$, since $P_1^\star$ is a relaxation of P_1'. Next, for any $x \in P_1^\star(\epsilon_1^\star)$ and any $S \subseteq N$, we have the following inequalities (where C_S denotes a minimum vertex cover on the subgraph induced by S).

$$\begin{aligned} f(S) - x(S) &= \sum_{v \in C_S} w_v - \sum_{i \in S} x_i \\ &\ge \sum_{v \in C_S} \Big(w_v - \sum_{i \in S:\, v \in i} x_i \Big) \\ &\ge \sum_{v \in C_S} \Big(w_v - \sum_{i \in N:\, v \in i} x_i \Big) \\ &\ge \sum_{v \in C_S} \Big(f(E(v)) - \sum_{i \in N:\, v \in i} x_i \Big) \\ &\ge \epsilon_1^\star \cdot |C_S| \ge \epsilon_1^\star \end{aligned}$$

where the last inequality follows from Theorem 4, which says the core is nonempty and thus $\epsilon_1^\star \ge 0$. Therefore, by the definition of ϵ_1', we have $\epsilon_1' \ge \epsilon_1^\star$. Thus, $\epsilon_1' = \epsilon_1^\star$.

Next we show that $P_1'(\epsilon_1') = P_1^\star(\epsilon_1^\star)$. First, it can be seen that $P_1'(\epsilon') \subseteq P_1^\star(\epsilon_1^\star)$, since $\epsilon_1' = \epsilon_1^\star$ and $P_1'(\epsilon')$ is a relaxation of $P_1^\star(\epsilon_1^\star)$. On the other hand, since for any $x \in P_1^\star(\epsilon_1^\star)$, x also satisfies the constraints of P_1' and gives the optimum. Therefore, $x \in P_1'(\epsilon_1')$; this implies that $P_1^\star(\epsilon_1^\star) \subseteq P_1'(\epsilon_1')$. □

Note that the above lemma also holds for fractional vertex cover games on non-bipartite graphs. In addition, in the above proof, if we let $S = N$, then $f(N) - x(N) \ge \epsilon_1^\star$. Since $f(N) = x(N)$ and $\epsilon_1^\star \ge 0$, we have $\epsilon_1^\star = 0$. This further implies that all inequalities in the above proof of $f(S) - x(S) \ge \epsilon_1^\star$ are tight. Thus, for any vertex v in a minimum vertex cover of G, $w_v - \sum_{i \in N:\, v \in i} x_i = 0$ and $E(v) \in fix(P_1'(\epsilon_1'))$.

Computing the Nucleolus

Having obtained a characterization for the least core, we next consider how to maximize the minimum excess among unfixed coalitions efficiently. The basic idea here is that the excess of a coalition can be decomposed according to vertices with their neighborhoods.

Given an allocation vector x, we first identify all neighborhoods of a vertex which may possibly have the minimum excess. For any $v \in V$, define

$$\mathcal{T}(v) = \{E(v)\} \cup \bigcup_{i \in E(v)} \{i\} \cup \bigcup_{i \in E(v)} \{E(v) - \{i\}\}.$$

Note that $\mathcal{T}(v)$ is the collection of subsets of $E(v)$, including all those with size 1, $|E(v)| - 1$, or $|E(v)|$. After kth iteration, we remove all fixed subsets from $\mathcal{T}(v)$; denote the resulting collection by $\mathcal{S}_k(v)$. That is,

$$\mathcal{S}_k(v) = \mathcal{T}(v) - fix(P_k'(\epsilon_k')).$$

An important property of $\mathcal{S}_k(v)$ it that it includes a subset which gives the smallest possible excess among all unfixed subsets of $E(v)$, shown by the following lemma.

Lemma 5. *Let x be any allocation vector. For any $v \in V$, there exists $\bar{S} \in \mathcal{S}_k(v)$ such that $f(S) - x(S) \ge f(\bar{S}) - x(\bar{S})$, for any $S \subseteq E(v)$ with $S \notin fix(P_k'(\epsilon_k'))$.*

Proof. Consider any $S \subseteq E(v)$ with $S \notin fix(P_k'(\epsilon_k'))$. Let $V_S = V(S) - \{v\}$. Then $f(S) = \min\{w_v, w(V_S)\}$. We consider the two cases.

- If $w_v \le w(V_S)$, then $f(S) - x(S) = w_v - x(S)$. If $E(v) \in \mathcal{S}_k(v)$, since $w_v \le w(V_S) \le w(V_{E(v)})$, we have
  $$w_v - x(S) \ge w_v - x(E(v)) = f(E(v)) - x(E(v)).$$
  Thus, $E(v)$ gives the desired $\bar{S}$.

  Otherwise, then we have $E(v) \in fix(P_k'(\epsilon_k'))$. Since $S \notin fix(P_k'(\epsilon_k'))$, there exists an edge $\bar{e} \in E(v) - S$ such that $\{\bar{e}\}$ is not fixed. Let $\bar{S} = E(v) - \{\bar{e}\}$; note that $\bar{S} \in \mathcal{S}_k(v)$ and $S \subseteq \bar{S}$. Since $w_v \le w(V_S) \le w(V_{\bar{S}})$, we have
  $$f(S) - x(S) = w_v - x(S) \ge w_v - x(\bar{S}) = f(\bar{S}) - x(\bar{S}).$$
- If $w_v > w(V_S)$, then
  $$f(S) - x(S) = \sum_{i=(v,v') \in S} (w_{v'} - x_i).$$
  Let $\bar{S} = \{\bar{i}\}$, where $\bar{i}$ is an unfixed edge in S (note that such an edge always exists as S is not fixed). Then $f(S) - x(S) \ge f(\bar{S}) - x(\bar{S})$ as all components in the above summation is non-negative.

This completes the proof. □

During the iterative procedure, after k iterations, define $T_k = \{v \in V \mid E(v) \in fix(P_k'(\epsilon_k'))\}$ to be the set of vertices for which their adjacent edges are fixed, and define $N_k = fix(P_k'(\epsilon_k')) \cap N$ to be the set of agents whose allocations have been fixed, denoted by α_i. Note that both T_k and N_k can be constructed in polynomial time.

Now we consider the following linear program.

$$\begin{aligned} (P_k^\star) \quad & \max \ \epsilon \\ & \text{s.t.} \quad x(S) + \epsilon \le f(S), \ \forall\, S \in \mathcal{S}_{k-1}(v), \ \forall\, v \in V \\ & \qquad\ \ x_i = \alpha_i, \ \forall\, i \in N_{k-1} \\ & \qquad\ \ x(E(v)) = f(E(v)), \ \forall\, v \in T_{k-1} \\ & \qquad\ \ x(N) = f(N) \\ & \qquad\ \ x_i \ge 0, \ \forall\, i \in N \end{aligned}$$

Lemma 6. *For any k, $\epsilon_k' = \epsilon_k^\star$ and $P_k'(\epsilon_k') = P_k^\star(\epsilon_k^\star)$, where ϵ_k' and $\epsilon_k^\star$ are the optimum values of $P_k'(\epsilon_k')$ and $P_k^\star(\epsilon_k^\star)$, respectively.*

Proof. We prove by induction. The basis case $k = 1$ follows from Lemma 4. Assume that the lemma holds for values less than k. We next prove the claim for the value k.

We first show that $P_k^\star$ is a relaxation of P_k' (note that this implies that $\epsilon_k' \le \epsilon_k^\star$). Consider any (x, ϵ) that satisfies P_k'. For any S fixed by $P_{k-1}'(\epsilon_{k-1}')$, $x(S)$ has a fixed value. By the induction hypothesis, $P_{k-1}'(\epsilon_{k-1}') = P_{k-1}^\star(\epsilon_{k-1}^\star)$; thus, $fix(P_{k-1}'(\epsilon_{k-1}')) = fix(P_{k-1}^\star(\epsilon_{k-1}^\star))$. Hence, N_{k-1} and $\bigcup_{v \in T_{k-1}} E(v)$ are in $fix(P_{k-1}^\star(\epsilon_{k-1}^\star))$; this implies that (x, ϵ) satisfies the equality constraints of $P_k^\star$. For any $v \in V$ and $S \in \mathcal{S}_{k-1}(v)$, since $S \notin fix(P_{k-1}^\star(\epsilon_{k-1}^\star))$, (x, ϵ) also satisfies the inequality constraints of $P_k^\star$.

Second, we show that for any $x \in P_k^\star(\epsilon_k^\star)$, $x \in P_{k-1}^\star(\epsilon_{k-1}^\star)$, i.e., $(x, \epsilon_{k-1}^\star)$ satisfies $P_{k-1}^\star$. If $k = 2$, as $\epsilon_1^\star = 0$ according to Lemma 4, we can see that x is also a core allocation. If $k > 2$, since $P'_{k-1}(\epsilon'_{k-1}) \subseteq P'_{k-2}(\epsilon'_{k-2})$, we have $fix(P'_{k-2}(\epsilon'_{k-2})) \subseteq fix(P'_{k-1}(\epsilon'_{k-1}))$. Thus, $N_{k-2} \subseteq N_{k-1}$ and $T_{k-2} \subseteq T_{k-1}$, and the equality constraints are satisfied. For the inequality constraints, consider any vertex $v \in V$ and any set $S \in \mathcal{S}_{k-2}(v)$.

- If $S \in \mathcal{S}_{k-1}(v)$, then $f(S) - x(S) \geq \epsilon^\star_k$. Since $P^\star_k$ is a relaxation of P'_k, we have $\epsilon^\star_k \geq \epsilon'_k \geq \epsilon'_{k-1} = \epsilon^\star_{k-1}$. Hence, $f(S) - x(S) \geq \epsilon^\star_{k-1}$.
- Otherwise, $S \notin \mathcal{S}_{k-1}(v)$. By the definition of $\mathcal{S}_{k-2}(v)$ and $\mathcal{S}_{k-1}(v)$, we know that $x(S)$ is newly fixed in $P'_{k-1}(\epsilon'_{k-1})$, which is $P^\star_{k-1}(\epsilon^\star_{k-1})$ by the hypothesis assumption. Thus, $f(S) - x(S) = \epsilon^\star_{k-1}$.

Hence, the inequality constraints are also satisfied.

We next show that for any optimal solution x of $P^\star_k$ which yields the optimum value $\epsilon^\star_k$ and any $S \subseteq N$ with $S \notin fix(P'_{k-1}(\epsilon'_{k-1}))$, $f(S) - x(S) \geq \epsilon^\star_k$. Let C be a minimum vertex cover of the subgraph induced by S. Then,

$$f(S) - x(S) \quad = \quad \sum_{v \in C} w_v - \sum_{i \in S} x_i = \sum_{v \in C} \Big(w_v - \sum_{i \in S:\, v \in i} x_i \Big).$$

If for any $v \in C$, $\{i \in S \mid v \in i\}$ is fixed by $P'_{k-1}(\epsilon'_{k-1})$, then S is also fixed by $P'_{k-1}(\epsilon'_{k-1})$, a contradiction. Hence, there exists $v \in C$ such that $\{i \in S \mid v \in i\}$ is not fixed by $P'_{k-1}(\epsilon'_{k-1})$, then by Lemma 5, there exists $\bar{S} \in \mathcal{S}_{k-1}(v)$ such that $f(S) - x(S) \geq f(\bar{S}) - x(\bar{S})$, which is at least $\epsilon^\star_k$.

Given the above analysis, we are ready to finish the inductive step. For any $x \in P^\star_k(\epsilon^\star_k)$, we have $x \in P^\star_k(\epsilon^\star_k) \subseteq P^\star_{k-1}(\epsilon^\star_{k-1}) = P'_{k-1}(\epsilon'_{k-1})$, As for any $S \subseteq N$ that is not fixed by $P'_{k-1}(\epsilon'_{k-1})$, $f(S) - x(S) \geq \epsilon^\star_k$, we know that $(x, \epsilon^\star_k)$ satisfies the constraints of P'_k, i.e., $x \in P'_k(\epsilon'_k)$ and $P^\star_k(\epsilon^\star_k) \subseteq P'_k(\epsilon'_k)$. This further implies that $\epsilon^\star_k \leq \epsilon'_k$, and thus, $\epsilon^\star_k = \epsilon'_k$. The proof of $P'_k(\epsilon'_k) \subseteq P^\star_k(\epsilon^\star_k)$ follows by a similar argument. This finishes the proof. □

Note that each program $P^\star_k$ has polynomial size, and the whole iterations terminate within $|N|$ programs. Therefore, we have the following theorem.

Theorem 5. *The nucleolus of vertex cover games on weighted bipartite graphs are in polynomial time solvable.*

Conclusions

We study fractional matching and vertex cover games, and give efficient algorithms to compute the least core and nucleolus. These algorithms can be respectively generalized to computing the nucleolus of the following two games:

- Fractional edge cover games on general weighted graphs: Agents are vertices and $f(S)$ of a coalition S is the cost of a minimum fractional edge cover of S.
- Clique (or equivalently, independent set) games on weighted bipartite graphs: Agents are edges and $f(S)$ is the value of a maximum bipartite complete graph of S.

All these games are combinatorial optimization games and have been studied in (Deng, Ibaraki, and Nagamochi 1999). A common feature of these games is that their cores are nonempty. Our algorithms build on this fact and the combinatorial structures of the problems. It is an interesting direction to explore the computation of the nucleolus of other combinatorial optimization games.

References

Aumann, R., and Maschler, M. 1985. Game theoretic analysis of a bankruptcy problem from the talmud. *J. of Economic Theory* 36:195–213.

Aziz, H.; Lachish, O.; Paterson, M.; and Savani, R. 2009. Wiretapping a hidden network. In *WINE*, 438–446.

Biro, P.; Kern, W.; and Paulusma, D. 2012. Computing solutions for matching games. *Interntional J. Game Theory* 41:75–90.

Branzei, R.; Inarra, E.; Tijs, S.; and Zarzuelo, J. 2006. A simple algorithm for the nucleolus of airport profit games. *Interntional J. Game Theory* 34:259–272.

Deng, X., and Papadimitriou, C. H. 1994. On the complexit of cooperative solution concepts. *Mathematics of Operations Research* 19:257–266.

Deng, X.; Fang, Q.; and Sun, X. 2009. Finding nucleolus of flow game. *J. Combinatorial Optimization* 18:64–86.

Deng, X.; Ibaraki, T.; and Nagamochi, H. 1999. Algorithmic aspects of the core of combinatorial optimization games. *Mathematics of Operations Research* 24:751–766.

Elkind, E., and Pasechnik, D. 2009. Computing the nucleolus of weighted voting games. In *SODA*, 327–335.

Elkind, E.; Goldberg, L.; Goldberg, P.; and Wooldridge, M. 2007. Computational complexity of weighted threshold games. In *AAAI*, 718–723.

Faigle, U.; Kern, W.; and Kuipers, J. 1998. Computing the nucleolus of min-cost spanning tree games is NP-hard. *International J. of Game Theory* 27:443–450.

Granot, D.; Maschler, M.; Owen, G.; and Zhu, W. R. 1996. The kernel/nucleolus of a standard tree game. *International J. of Game Theory* 25:219–244.

Kern, W., and Paulusma, D. 2003. Matching games: The least core and the nucleolus. *Mathematics of Operations Research* 28:294–308.

Kopelowitz, A. 1967. *Computation of the Kernels of Simple Games and the Nucleolus of n-Person Games*. RM-31, Hebrew University of Jerusalem.

Lemaire, J. 1984. An application of game theory: Cost allocation. *ASTIN Bulletin* 14:61–81.

Maschler, M.; Peleg, B.; and Shapley, L. S. 1979. Geometric properties of the kernel, nucleolus, and related solution concepts. *Mathematics of Operations Research* 4:303–338.

Megiddo, N. 1978. Computational complexity and the game theory approach to cost allocation for a tree. *Mathematics of Operations Research* 3:189–196.

Owen, G. 1975. On the core of linear production games. *Mathematical Programming* 9:358–370.

Schmeidler, D. 1969. The nucleolus of a characteristic function game. *SIAM J. Appl. Math.* 17:11631170.

Schrijver, A. 2003. *Combinatorial Optimization: Polyhedra and Efficiency*. Springer.

Shapley, L., and Shubik, M. 1971. The assignment game i: The core. *Interntional J. Game Theory* 1:111–130.

Solymosi, T., and Raghavan, T. 1994. An algorithm for finding the nucleolus of assignment games. *Interntional J. Game Theory* 23:119–143.