On the core and f-nucleolus of flow games

Walter Kern
Daniël Paulusma

Univerisity of Twente
Durham University

Bristol Algorithms Days 2008

Introduction

Game Theory: models and analyses situations of conflict involving a number of economic agents (the players) who take actions and make decisions.

Cooperative Game Theory: the players may choose to cooperate with each other in order to optimise their profits or costs.

A cooperative game is given by a set N of players and a value function $v : 2^N \longrightarrow \mathbb{R}$ with $v(\emptyset) = 0$.

A coalition is any subset $S \subseteq N$. We refer to $v(S)$ as the value of coalition S, i.e., the maximal profit or minimal cost that the players in S achieve by cooperating with each other.

The value $v(E)$ is called the total value of the game.

As in many games $v(N) \geq \sum v(S_i)$ holds for every partition $S_1 \cup \cdots \cup S_k$ of N, we often assume that the grand coalition N is formed.

The central problem is then to allocate the total value $v(N)$ to the individual players $i \in N$ in a "fair" way.

An allocation is a vector $x \in \mathbb{R}^N$ with $x(N) = v(N)$.

Here we adopted the standard notation $x(S) = \sum_{i \in S} x_i$.

A solution concept $\mathcal{S}$ prescribes a set $\mathcal{S}(N, v)$ of allocations for each game (N, v).

The choice for a specific solution concept $\mathcal{S}$ depends on the notion of fairness specified within the decision model and the complexity of computing an $\mathcal{S}$-allocation.

We are interested in the computational complexity of

1. computing an allocation in $\mathcal{S}(N, v)$
2. testing membership in $\mathcal{S}(N, v)$
3. testing nonemptiness of $\mathcal{S}(N, v)$

of solution concepts $\mathcal{S}$ for the class of flow games.

Flow games

The v-values of many cooperative games are derived from solving an underlying discrete optimization problem.

Flow games were introduced by Kalai & Zemel [1982].

The underlying structure of a flow game (N, v) is a (flow) network, i.e., a directed graph $G = (V, E)$ with source $s \in V$, sink $t \in V$ and positive edge capacities $c \in \mathbb{R}^E$.

Each player in this game owns exactly one arc, so $N = E$.

Players cooperate with each other in order to allow a flow going from s to t. So, the v-values are given by

$$v(A) := \text{maximum flow in } (V, A), \qquad A \subseteq E.$$

A simple flow game is a flow game on a network with unit capacities ($c_e = 1$ for all $e \in E$).

The core

The core of a game (N, v) is the set of allocations that are fair in the sense that every coalition S gets at least its value $v(S)$:

$$\begin{aligned} \text{core}(N, v): \quad & x(S) \geq v(S) \qquad S \in 2^N \setminus \{\emptyset, N\} \\ & x(N) = v(N) \end{aligned}$$

A coalition $S \subseteq N$ is essential if $v(S) > \sum v(S_i)$ holds for every non-trivial partition $S = S_1 \cup \ldots \cup S_k$. Clearly, we may replace 2^N by the set of essential coalitions.

Nevertheless, in general, the number of core inequalities remains exponential.

The core of a flow game

For a flow game (E, v), core allocations $x \in \text{core}(E, v)$ are easy to find (Kalai & Zemel [1982]).

For $S, T \subset V$ with $s \in S, t \in T$ and $S \cap T = \emptyset$ we write

$$[S:T] = \{(i,j) \in E \mid i \in S, j \in T\}.$$

We fix an arbitrary min $s - t$ cut $[S:T] \subseteq E$. Then

$$x_e := \begin{cases} c_e & \text{if } e \in [S:T] \\ 0 & \text{else} \end{cases}$$

satisfies the core constraints

$x(E) = v(E)$

$x(A) \geq v(A)$ for all $A \subseteq E$

due to the Max-Flow Min-Cut Theorem.

However, testing membership is NP-hard (Fang et al. [2002]).

The core of a simple flow game

Let $\mathcal{P} \subseteq 2^E$ denote the family of $s-t$ paths, i.e., directed paths starting in s and ending in t.

Observation

Let (E, v) be a simple flow game, then

$$\begin{array}{llr} core(E,v): & x(P) \geq 1 & P \in \mathcal{P} \\ & x_e \geq 0 & e \in E \\ & x(E) = v(E). & \end{array}$$

Proof.

Any $A \subseteq E$ contains $\ell = v(A)$ pairwise edge-disjoint paths $P_1, \ldots, P_\ell \in \mathcal{P}$. Hence the above constraints imply $x(A) \geq x(P_1) + \ldots + x(P_\ell) \geq \ell = v(A)$. □

Let $S, T \subset V$ with $s \in S, t \in T$ and $S \cap T = \emptyset$.

Recall that $[S : T] = \{(i, j) \in E \mid i \in S, j \in T\}$.

We write $[T : S] = \{(j, i) \in E \mid i \in S, j \in T\}$.

A min cut edge is an edge $e \in E$ that is contained in some min cut $[S : T] \subseteq E$. Let M denote the set of min cut edges.

A reverse edge is an edge $e \in [T : S]$ for some min cut $[S : T] \subseteq E$. Let R denotes the set of reverse edges.

Proposition

Let (E, v) be a simple flow game, then

$$\begin{array}{llll} core(E,v): & x(P) \geq 1 & & P \in \mathcal{P},\ P \cap R \neq \emptyset \\ & x(P) = 1 & & P \in \mathcal{P},\ P \cap R = \emptyset \\ & x_e & \geq 0 & e \in M \\ & x_e & = 0 & e \notin M. \end{array}$$

All inequalities can be satisfied strictly.

Let $D := E \backslash (M \cup R)$ denote the set of dummy edges.

Can we get rid of D?

Let $D := E \backslash (M \cup R)$ denote the set of dummy edges.

Can we get rid of D?

Yes, there is even more.

We define a max flow set in a simple network as a set of k pairwise edge-disjoint s-t paths $P_1, \ldots, P_k$ in $\mathcal{P}$ that form a max flow of value $k = v(E)$.

Lemma

Let (E, v) be a simple flow game. We may without loss of generality assume that (E, v) is minimal, i.e., $E = M \cup R$ and for any max flow set $\{P_1, \ldots, P_k\}$:

(i) $M = P_1 \cup \ldots \cup P_k$

(ii) $V = V(P_1) \cup \ldots \cup V(P_k)$.

Theorem

Let (E, v) be a minimal simple flow game. We can find (in polynomial time) sets $\mathcal{P}_0, \mathcal{P}_1 \subseteq \mathcal{P}$ of size $O(|E|^2)$ such that

$$
\begin{array}{llll}
core(E, v): & x(P) = 1 & & P \in \mathcal{P}_0 \\
 & x(P) \geq 1 & & P \in \mathcal{P}_1 \\
 & x_e \quad \geq 0 & & e \in M \\
 & x_e \quad = 0 & & e \in R.
\end{array}
$$

Two structural disadvantages of the notion of a core are:

1. The core of a game may be empty.
2. If $x(S) - v(S) < x(S') - v(S')$ then S' is more satisfied with x than S.

The core of a flow game is nonempty. However 2 is still true.

The nucleolus

Given an allocation $x \in \mathbb{R}^N$ for some game (N, v), we first define the excess of a nonempty coalition $S \subsetneq N$ as

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

By ordering all excesses into a non-decreasing sequence, we obtain the *excess vector* $\theta(x)$.

The *nucleolus* of (N, v) is defined as the set of allocations that lexicographically maximise $\theta(x)$ over the set of imputations, i.e., over all allocations x with $x_i \geq v(\{i\})$ for all $i \in N$.

Theorem (Schmeidler 1969)

If the nucleolus of a game is nonempty, then it is a singleton.

Due to Maschler, Peleg, & Shapley (1979), we can compute the nucleolus of a game (N, v) as follows.

For convenience, assume that core(N, v) is nonempty.

We first seek for an allocation $x \in \mathbb{R}^N$ satisfying all coalitions as much as possible by solving

$$
\begin{array}{llll}
(\mathrm{LP}_1) & \varepsilon_1 & := & \max \varepsilon \\
 & & & x(N) = v(N) \\
 & & & x(S) \geq v(S) + \varepsilon \quad S \in 2^N \backslash \{\emptyset, N\}.
\end{array}
$$

Note that $\varepsilon_1 \geq 0$ as core(N, v) is nonempty.

The set of allocations $x \in \mathbb{R}^N$ for which (x, ε_1) is an optimal solution of (LP_1) is known as the least core $P_1(\varepsilon_1)$ of (N, v).

Note that $P_1(\varepsilon_1)$ consists of those imputations that maximise the smallest excess. Hence it contains the nucleolus.

This idea is further pursued.

For a polyhedron $P \subseteq R^N$ let Fix P consists of all subsets $S \subseteq 2^N$ that are fixed by P, i.e., $x(S) = y(S)$ for all $x, y \in P$.

If not all coalitions are fixed we further increase $x(S)$ for all $S \notin \text{Fix } P(\varepsilon_1)$ etc.

This leads to a sequence of linear programs for $r \geq 2$

$$
\begin{array}{llll}
(\mathrm{LP}_r) & \varepsilon_r & := & \max \varepsilon \\
 & & & x \in P_{r-1}(\varepsilon_{r-1}) \\
 & & & x(S) \geq v(S) + \varepsilon \quad S \notin \text{Fix } P_{r-1}(\varepsilon_{r-1}),
\end{array}
$$

where $P_{r-1}(\varepsilon_{r-1})$ denotes the set of imputations $x \in \mathbb{R}^N$ for which (x, ε_{r-1}) is an optimal solution of (LP_{r-1}).

The dimension of the feasible regions of (LP_r) decreases in each step, so we end up with a unique optimal solution x^*, the nucleolus of (N, v), after at most $|N|$ iterations.

The nucleolus is in general difficult to compute due to the exponentially many constraints in each (LP_r).

The nucleolus of a simple flow game

Just as for the core we are able to eliminate many redundant $s - t$ path constraints such that only a polynomial number of these constraints remain.

Theorem

The nucleolus can be computed in polynomial time for the class of simple flow games.

This result has also been obtained by Deng, Fang & Sun [2006] and by Potters, Reijnierse, Biswas [2006].

Both papers use the ellipsoid method.

The f-nucleolus

The definition of the nucleolus requires that all coalitions have to satisfy the same type of additive constraints ("$+\varepsilon$").

In some situations, a multiplicative variant is more natural.

We define a priority function as a mapping $f : 2^N \to \mathbb{R}^+$ that gives priority $f(S)$ to coalition S.

We set $\varepsilon_0^f := 0$ and let $P_0(0)$ consist of all imputations.

For $r = 1, 2, \ldots,$ we let $P_r(\varepsilon_r^f)$ consist of all optimal solutions (x, ε_r^f) of the linear program

$$
(\mathrm{LP}_r^f)\ \ \varepsilon_r^f := \begin{array}{l} \max \varepsilon \\ x \in P_{r-1}(\varepsilon_{r-1}^f) \\ x(S) \geq v(S) + \varepsilon f(S) \ S \notin \mathrm{Fix}\ P_{r-1}(\varepsilon_{r-1}^f) \end{array}
$$

until ε_r^f is unbounded. Then the f-prenucleolus of (N, v) is the polyhedron $P_r(\varepsilon_r^f)$.

The f-nucleolus of flow games

We call a priority function f *suitable* if $f(S) > 0$ whenever $v(S) > 0$.

The f-nucleolus is called the

- nucleolus if $f(S) = 1$ for all nonempty $S \subset N$
- nucleon if $f(S) = v(S)$ for all nonempty $S \subset N$
- per-capita nucleolus if $f(S) = |S|$ for all nonempty $S \subset N$.

Computing the nucleolus is NP-hard for the class of flow games (Deng, Fang & Sun [2006]).

We generalize as follows.

Theorem

For any suitable priority function f, computing an allocation in the f-nucleolus is NP-hard for the class of flow games.

For simple flow games we can eliminate $s - t$ path constraints such that only a polynomial number of these constraints remain.

Theorem

An allocation in the nucleon can be computed in polynomial time for the class of simple flow games.

Open problems

1. Is computing the f-nucleolus of a simple flow game polynomially solvable for all suitable priority functions f?
2. Can the class of simple flow games be extended to a larger subclass of flow games for which efficient algorithms exist for computing the core, nucleolus, or nucleon?

Answering question 1 might not be easy. Although the proofs for computing the nucleolus and nucleon of a simple flow game are based on the same method, the $s - t$ path constraints we may remove are different in each case.

A candidate for an answer to question 2 might be the subclass of flow games that are defined on a network $G = (V, E)$ with edge capacities $c(u, v) = c(u) + c(v)$ for vertex capacities $c \in \mathbb{R}^V_+$.