Monotone cooperative games and their threshold versions

Haris Aziz Felix Brandt Paul Harrenstein

Ludwig-Maximilians-Universität München

COST-ADT COMSOC School, April 13, 2010

PREFERENCE AGGREGATION IN MULTIAGENT SYSTEMS

Cooperative game theory

- General idea of power indices:

  "If a player contributes more to the values of the coalitions, it should get more payoff."

- This talk concentrates more on *stability* aspect of payoff distribution.
- Stable and fair resource allocation is an important issue in networks, distributed systems, operations research and multiagent systems.

TU Cooperative games

***TU cooperative game*:**

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

TU Cooperative games

***TU cooperative game*:**

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

***Simple game*:**

- A ***simple game*** is a monotone cooperative game (N, v) with $v : 2^N \to \{0, 1\}$ such that $v(\emptyset) = 0$ and $v(N) = 1$.
- A coalition $S \subseteq N$ is *winning* if $v(S) = 1$ and *losing* if $v(S) = 0$.

TU Cooperative games

***TU cooperative game*:**

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

***Simple game*:**

- A ***simple game*** is a monotone cooperative game (N, v) with $v : 2^N \to \{0, 1\}$ such that $v(\emptyset) = 0$ and $v(N) = 1$.
- A coalition $S \subseteq N$ is *winning* if $v(S) = 1$ and *losing* if $v(S) = 0$.

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

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

Goal

- Examine classes of monotone cooperative games and their threshold versions.
- Complexity of core related solutions of monotone cooperative games.
- Complexity of computing the smallest winning coalition for simple games.

Monotone cooperative game classes

A ***weighted voting game (WVG)***$[q; w_1, \ldots, w_n]$ is a simple game (N, v) for which there is a quota $q \in \mathbb{R}^+$ and a *weight* w_i for each player i such that

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

A ***multiple weighted voting game*** **(MWVG)** is the simple game (N, v) for which there are WVGs $(N, v_1), \ldots, (N, v_m)$ such that S is winning if and only if S is winning in each of the constituent WVGs.

Monotone cooperative games classes

Spanning connectivity game (SCG)*:** For each connected undirected graph (V, E), the ***spanning connectivity game (SCG) is the simple game (N, v) where

- $N = E$
- S is winning if and only if S is a connected spanning subgraph.

***Simple coalitional skill game (SCSG)*:**

- Let $N = \{1, \ldots, n\}$ is the set of player and $\Sigma = \{\sigma_1, \ldots, \sigma_k\}$ be the set of skills, s.t. each player has a set of skills $\Sigma_i \subseteq \Sigma$.
- The ***simple coalitional skill game (SCSG)*** is a simple game in which a coalition S is winning if and only if for each skill in Σ, at least one player in S has that skill.

Monotone cooperative games classes

Matching game*:** Let $G = (V, E, w)$ be a weighted undirected graph. The ***matching game corresponding to G is the cooperative game (N, v) with

- $N = V$
- for each $S \subseteq N$, the value $v(S)$ equals the weight of the maximum weighted matching of the subgraph induced by S.

Monotone cooperative games classes

Matching game: Let $G = (V, E, w)$ be a weighted undirected graph. The ***matching game*** corresponding to G is the cooperative game (N, v) with

- $N = V$
- for each $S \subseteq N$, the value $v(S)$ equals the weight of the maximum weighted matching of the subgraph induced by S.

Graph game (GG): The ***graph game (GG)*** has a similar setting as matching games but here, for $S \subseteq N$, $v(S)$ is the sum of the weight of edges in the subgraph induced by S.

Monotone cooperative games classes

Matching game*:** Let $G = (V, E, w)$ be a weighted undirected graph. The ***matching game corresponding to G is the cooperative game (N, v) with

- $N = V$
- for each $S \subseteq N$, the value $v(S)$ equals the weight of the maximum weighted matching of the subgraph induced by S.

Graph game (GG)*:** The ***graph game (GG) has a similar setting as matching games but here, for $S \subseteq N$, $v(S)$ is the sum of the weight of edges in the subgraph induced by S.

Network flow game (NFG): For a flow network (V, E, c, s, t), the associated ***network flow game (NFG)*** is the cooperative game (N, v),

- $N = E$
- for each $S \subseteq E$ the value $v(S)$ is the value of the maximum flow f restricted to edges in S

Solution concepts: core

A solution concept associates with each cooperative game (N, v) a set of *payoff vectors* $(x_1, \ldots, x_n) \in \mathbb{R}^N$ such that $\sum_{i \in N} x_i = v(N)$, where x_i denotes player i's share of $v(N)$.
Notation: $x(S) = \sum_{i \in S} x_i$

> $v(N)$ *is the amount which the players can earn if they work together.*
> *The aim is to divide* $v(N)$ *among the players in a stable manner.*

Solution concepts: core

A solution concept associates with each cooperative game (N, v) a set of *payoff vectors* $(x_1, \ldots, x_n) \in \mathbb{R}^N$ such that $\sum_{i \in N} x_i = v(N)$, where x_i denotes player *i*'s share of $v(N)$.

Notation: $x(S) = \sum_{i \in S} x_i$

> $v(N)$ *is the amount which the players can earn if they work together.*
> *The aim is to divide* $v(N)$ *among the players in a stable manner.*

Core: A payoff vector $x = (x_1, \ldots, x_n)$ is in the ***core*** of a cooperative game (N, v) if for all $S \subset N$, $x(S) \geq v(S)$,

Solution concepts: core

A solution concept associates with each cooperative game (N, v) a set of *payoff vectors* $(x_1, \ldots, x_n) \in \mathbb{R}^N$ such that $\sum_{i \in N} x_i = v(N)$, where x_i denotes player *i*'s share of $v(N)$.

Notation: $x(S) = \sum_{i \in S} x_i$

> *$v(N)$ is the amount which the players can earn if they work together.*
> *The aim is to divide $v(N)$ among the players in a stable manner.*

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

Solution concepts: core

A solution concept associates with each cooperative game (N, v) a set of *payoff vectors* $(x_1, \ldots, x_n) \in \mathbb{R}^N$ such that $\sum_{i \in N} x_i = v(N)$, where x_i denotes player i's share of $v(N)$.
Notation: $x(S) = \sum_{i \in S} x_i$

$v(N)$ is the amount which the players can earn if they work together.
The aim is to divide $v(N)$ among the players in a stable manner.

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

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

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

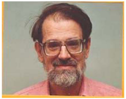
.

Solution concepts: least core

- For $\epsilon > 0$, a payoff vector vector x is in the **ϵ-core** if for all $S \subset N$, $e(x, S) \geq -\epsilon$.
- The ***least core*** is the refinement of the ϵ-core and is the solution of the following LP:

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

Introduced in [Shapley and Shubik, Econometrica, 1966]

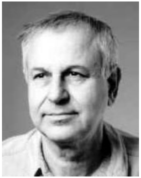

Lloyd Shapley


Martin Shubik

Solution concepts: nucleolus

The **nucleolus** is a lexicographical refinement of the least core.

Introduced in [Schmeidler, SIAM J of App. Math., 1969]



Solution concepts: CoS

Definition

- For a given coalitional game $G = (N, v)$ and a payment $\triangle \in \mathbb{R}^+$, the *adjusted coalitional game* $G(\triangle) = (N, v')$ is exactly like (N, v) except that $v'(N) = v(N) + \triangle$.
- The ***cost of stability (CoS)*** of a game is the minimum supplemental payment $CoS(G)$ such that $G(CoS(G))$ has a nonempty core. $CoS(G)$ is the solution of the following LP:

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

Solution concepts: CoS

Definition

- For a given coalitional game $G = (N, v)$ and a payment $\triangle \in \mathbb{R}^+$, the *adjusted coalitional game* $G(\triangle) = (N, v')$ is exactly like (N, v) except that $v'(N) = v(N) + \triangle$.
- The ***cost of stability (CoS)*** of a game is the minimum supplemental payment $CoS(G)$ such that $G(CoS(G))$ has a nonempty core. $CoS(G)$ is the solution of the following LP:

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

[Bachrach, Meir, Zuckerman, Rothe and Rosenschein. The cost of stability in weighted voting games. In AAMAS 2009]

Computational Problems

For any solution concept $X \in \{$ least core, nucleolus, ϵ-core$\}$, we consider the following standard computational problems:

- **IN-**X: given a cooperative game (N, v) and payoff vector p, check whether p is in solution X of (N, v).
- **CONSTRUCT-**X: given a cooperative game (N, v), compute a payoff vector p, which is in solution X of (N, v).

Computational Problems

For any solution concept $X \in \{$ least core, nucleolus, ϵ-core$\}$, we consider the following standard computational problems:

- **IN-**X: given a cooperative game (N, v) and payoff vector p, check whether p is in solution X of (N, v).
- **CONSTRUCT-**X: given a cooperative game (N, v), compute a payoff vector p, which is in solution X of (N, v).
- **CoS**: given a cooperative game (N, v), compute $CoS((N, v))$.

Computing the smallest winning coalition

- The ***length*** of a simple game is the size of the smallest winning coalition.
- LENGTH: For a simple game (N, v), compute the smallest winning coalition.
- *“What is the minimum number of players needed to get the job done?”*

Game class	Complexity of LENGTH
WVG	*P*
T-Matching	*P*
T-NFG	*NP*-hard
MWVG	*NP*-hard
SCSG	*NP*-hard
T-GG$^+$	*NP*-hard

Table: Complexity of LENGTH

Computing the smallest winning coalition

- The ***length*** of a simple game is the size of the smallest winning coalition.
- LENGTH: For a simple game (N, v), compute the smallest winning coalition.
- *"What are the minimum number of players needed to get the job done?"*

Game class	Complexity of LENGTH
WVG	*P*
T-Matching	*P*
T-NFG	*NP*-hard
MWVG	*NP*-hard
SCSG	*NP*-hard
T-GG$^+$	*NP*-hard

Table: Complexity of LENGTH

Computing the smallest winning coalition

Proposition

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

Proof idea

- *Main idea: Reduction of the problem to computing maximum weighted matchings of at most $\lfloor |V|/2 \rfloor$ different transformed graphs.*
- *Suppose we want to compute the maximum matching of size s of $G = (V, E, w)$. Then transform graph G into G' by creating $j = |V| - 2s$ new nodes $V' = \{v'_1, \dots, v'_j\}$ and joining each node in V' to each node in V with an edge of weight $W = \sum_{i=1}^{|E|} w(e_i)$.*
- *Let M' be the maximum (perfect) matching of G'. Then $M = M' \cap E$ is the maximum matching of G with size s.*

Core related solutions of cooperative games

	least core	CoS	nucleolus
GG^+			
SCG			
SCSG			
NFG			
Matching			
WVG			
T-Matching			
T-NFG			
T-GG^+			
MWVG			

Table: Complexity of monotone cooperative games

Core related solutions of cooperative games

	least core	CoS	nucleolus
GG^+	*P*[2]	*P*[2]	*P*[2]
SCG	*P*	*P*	*P* [1]
SCSG (fixed #skills)	*P*	*P*	*P*
NFG	*P* [4]	*P*[4]	?
Matching	*P* [5]	*P*	?
WVG	*NP*-hard [3]	*NP*-hard [3]	*NP*-hard [3]
T-Matching	*NP*-hard	*NP*-hard	*NP*-hard
T-NFG	*NP*-hard [6]	*NP*-hard [6]	*NP*-hard
T-GG^+	*NP*-hard	*NP*-hard	*NP*-hard

Table: Complexity of monotone cooperative games

Core related solutions of cooperative games

	least core	CoS	nucleolus
GG^+	*P*[2]	*P*[2]	*P*[2]
SCG	*P*	*P*	*P* [1]
SCSG (fixed #skills)	*P*	*P*	*P*
NFG	*P* [4]	*P*[4]	?
Matching	*P* [5]	*P*	?
WVG	*NP*-hard [3]	*NP*-hard [3]	*NP*-hard [3]
T-Matching	*NP*-hard	*NP*-hard	*NP*-hard
T-NFG	*NP*-hard [6]	*NP*-hard [6]	*NP*-hard
T-GG^+	*NP*-hard	*NP*-hard	*NP*-hard

Table: Complexity of monotone cooperative games

CoS of Matching games

Proposition

For matching games, there exists a polynomial-time algorithm to compute CoS.

Proof idea

- *Idea: use ellipsoid method and construct a polynomial time separation oracle.*

  > *If one can decide feasibility of LPs in polynomial time then one can compute optimal solutions in polynomial time.*

- *For payoff, $x = (x_1, \ldots, x_n)$ and $\epsilon > 0$, returns "yes" if the minimum excess of G with respect to x is more than $-\epsilon$ and otherwise returns the violated constraint.*
- *For a payoff vector x and $G = (N, E, w)$, the graph G'_x is (N, E, w'), where for each edge (i, j), $w'((i, j)) = w((i, j)) - x_i - x_j$.*
- *For any coalition S, $-e(x, S)$ is equal to the weight of a maximum matching of G'_x restricted to nodes in S.*

CoS of SCSGs

Proposition

For a SCSG with a constant number of skills, the CoS can be computed in polynomial time.

Proof idea

- *Reduce the SCSG with n players and k skills into a MWVG with n players and k constituent WVGs, each with quota one and weights zero or one.*
- *Consider SCSG (N, v) with n players and k skills. Then for $j = 1, \ldots, k$ and for each skill σ_j, construct a corresponding WVG $(N, v_j) = [q^j; w_1^j, \ldots, w_n^j]$ where $q^j = 1$ and for $i = 1, \ldots, n$, $w_i^j = 1$ if i has skill s_j and zero otherwise.*
- *In [Elkind and Pasechnik, SODA 2009], an algorithm was presented which computes the nucleolus of a MWVG which is polynomial in n and the sum of the weights of the WVGs.*
- *Reduce our separation oracle to a subroutine in [Elkind and Pasechnik, SODA 2009]*

Cooperative games

Proposition

If computing the length of a simple game (N, v) is NP-hard, then IN-ϵ-CORE for (N, v) is NP-hard.

(Applies for e.g. to T-NFG and T-GG$^+$)

Observation

If IN-ϵ-CORE is NP-hard and unless $P = NP$, then there is no polynomial time separation oracle to solve the least core LP or the CoS LP.

(Means that we need some efficient combinatorial algorithm to compute the least core payoff vectors)

Computing the least core minimum excess

Deng and Fang [Algorithmic cooperative game theory. In Pareto Optimality, Game Theory And Equilibria, 2008] note that

> *"the most natural problem is how to efficiently compute the value* ϵ_1 *for a given cooperative game. The catch is that the computation of* ϵ_1 *requires one to solve a linear program with [an] exponential number of constraints."*

Computing the least core minimum excess

Deng and Fang [Algorithmic cooperative game theory. In Pareto Optimality, Game Theory And Equilibria, 2008] note that

> *"the most natural problem is how to efficiently compute the value ϵ_1 for a given cooperative game. The catch is that the computation of ϵ_1 requires one to solve a linear program with [an] exponential number of constraints."*

It is not clear whether the least core minimum excess can be computed efficiently even if a least core payoff vector is given.

Computing the least core minimum excess

Deng and Fang [Algorithmic cooperative game theory. In Pareto Optimality, Game Theory And Equilibria, 2008] note that

> *"the most natural problem is how to efficiently compute the value ϵ_1 for a given cooperative game. The catch is that the computation of ϵ_1 requires one to solve a linear program with [an] exponential number of constraints."*

It is not clear whether the least core minimum excess can be computed efficiently even if a least core payoff vector is given.

Proposition

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

Passer-consistent representation: The representation can easily extend a game to one with one more player which is a passer. (WVGs, MWVGs, SCSGs etc.)

Structure of least core payoffs

Proposition

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

Proposition

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

Summary

	least core	CoS	nucleolus
GG^{+}	*P*[2]	*P*[2]	*P*[2]
SCG	*P*	*P*	*P* [1]
SCSG (fixed #skills)	*P*	*P*	*P*
NFG	*P* [4]	*P*[4]	?
Matching	*P* [5]	*P*	?
WVG	*NP*-hard [3]	*NP*-hard [3]	*NP*-hard [3]
T-Matching	*NP*-hard	*NP*-hard	*NP*-hard
T-NFG	*NP*-hard [6]	*NP*-hard [6]	*NP*-hard
T-GG^{+}	*NP*-hard	*NP*-hard	*NP*-hard

Table: Complexity of monotone cooperative games

Conclusions

***Summary*:**

- Complexity of finding the smallest winning coalition of many simple games.
- Complexity of core related questions for many games.
- Threshold versions are not only less expressive but also seem to be harder to handle computationally.
- Structure of least core payoffs.

***New or open questions*:**

- Is the complexity of CoS and the least core same? Can one problem reduce to another?
- Find the CoS bounds for classes of games.
- The complexity of nucleolus of matching games and network flow games are longstanding open problems.

Summary

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

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

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

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

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

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