Cooperative Games

Lecture 9: Representation and Complexitity issues

Stéphane Airiau

ILLC - University of Amsterdam

Today

- Can we efficiently compute a solution?
- Compact representation

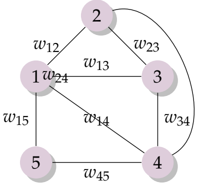
Representation by enumeration

- Let us assume we want to write a program for computing a solution concept.
- How do we represent the input of a TU game?
- Straighforward representation by enumeration requires **exponential space**.
- Brute force approach may appear good as complexity is measured in term of the **input size**.
- we need **compact** or **succinct** representation of coalitional games.
- e.g., a representation so that the input size is a polynomial in the number of agents.
- In general, the more succinct a representation is, the harder it is to compute, hence we look for a balance between succinctness and tractability.

Induced subgraph games

An induced subgraph game is a coalitional game defined by an undirected weighted graph $\mathcal{G} = (V,W)$ where V is the set of vertices and $W : V \to V$ is the set of edges weights. For $(i,j) \in V^2$, w_{ij} is the weight of the edge between i and j.

- $N = V$, i.e., each agent is a node in the graph.
- for all $\mathcal{C} \subseteq N$, $v(\mathcal{C}) = \sum_{(i,j)\in\mathcal{C}} w_{ij}$.



It is a **succinct** representation: using an adjacency matrix, we need to provide n^2 entries.
However, it is **not complete**. Some TU games cannot be represented by a induced subgraph game (e.g., a majority voting game).

Proposition

Let (V,W) be a induced subgraph game. If all the weights are nonnegative then the game is convex.

Proof

$$\begin{aligned} v(S)+v(T) &= \sum_{(i,j)\in S^2} w_{ij} + \sum_{(i,j)\in T^2} w_{ij} = \sum_{(i,j)\in S^2 \vee (i,j)\in T^2} w_{ij} + \sum_{(i,j)\in (S\cap T)^2} w_{ij} \\ &\leqslant \sum_{(i,j)\in (S\cup T)^2} w_{ij} + \sum_{(i,j)\in (S\cap T)^2} w_{ij} = v(S\cup T) + v(S\cap T) \end{aligned}$$

□

Proposition

Let (V,W) be a induced subgraph game. If all the weights are nonnegative then membership of a payoff vector in the core can be tested in polynomial time.

Theorem

Let (V,W) an induced subgraph game. The Shapley value of an agent $i \in V$ is $Sh_i(N,v) = \frac{1}{2} \sum_{(i,j)\in N^2 \mid i\neq j} w_{ij}$.

The Shapley value can be computed in $O(n^2)$ time.

Proof

Let (V,W) a induced subgraph game. We can view this game as the sum of the following $|W|$ games (i.e., one game per edge): $G_{ij} = (V, \{w_{ij}\})$, $(i,j) \in V^2$.
For a game G_{ij}, i and j are substitutes, and all other agents $k \neq i,j$ are dummy agents. Using the symmetry axiom, $Sh_i(G_{ij}) = Sh_j(G_{ij})$. Using the dummy axiom, $Sh_k(G_{ij}) = 0$.
Hence, $Sh_i(G_{ij}) = \frac{1}{2} w_{ij}$.
Since (V,W) is the sum of all two-player games, by the additivity axiom, $Sh_k = \sum_{i,j} Sh_k(G_{ij}) = \sum_{k,i} w_{ij}$ □

Theorem

Let (V,W) be an induced subgraph game. Testing the nonemptyness of the core is NP-complete.

Other results

- Induced subgraph games may not be monotone (because of negative weights).
- Efficient network-flow based algorithm for checking whether an outcome is in the core (when weights are positive) [Deng & Papadimitriou, 94]
- If weights can be negative,
  - checking whether the core is empty is NP-complete.
  - checking whether an outcome is in the core is coNP-complete [Deng & Papadimitriou, 94]
- checking whether an outcome is
  - in the kernel is Σ_2^p-complete.
  - in the bargaining set is Π_2^p-complete [Greco & al, 2011]

Network flow games

- Players are edges of a network with a source s and a sink t.
- Each edge e has a capacity $c_e \in \mathbb{R}$, indicating how much it can carry
- The value of a colition $\mathcal{C} \subset N$ is the value of the maximum amount of flow that can be sent from s to t using only the edges in C.

Solution concepts studied by Granot and Granot 92.
Computing nucleolus is easy when weights are one, hard otherwise [Deng 09]
Variant: can the network carry a flow of at least k? [Bachrach et al. 09]

A representation for superadditive games

Instead of storing a value for each coalition, we can store the positive synergies between agents.
Let (N,v) be a superadditive game and (N,s) its **synergy representation**. Then for a coalition $\mathcal{C} \subseteq N$,

$$v(\mathcal{C}) = \left(\max_{\{\mathcal{C}_1,\mathcal{C}_2,\ldots,\mathcal{C}_k\} \in \mathscr{S}_\mathcal{C}} \sum_{i=1}^{k} v(\mathcal{C}_i) \right),$$

where $\mathscr{S}_\mathcal{C}$ is the set of all partition of $\mathcal{C}$.

Example: $N = \{1,2,3\}$, $v(\{i\}) = 1$, $v(\{1,2\}) = 3$, $v(\{1,3\}) = 2$, $v(\{2,3\}) = 2$, $v(\{1,2,3\}) = 4$.
We can represent this game by $v(\{i\}) = 1$, $v(\{1,2\}) = 3$.

This representation may still require a space exponential in the number of agents, but for many games, the space required is much less.

Theorem

It is NP-complete to determine the value of some coalitions for a coalitional game specified with the synergy representation. In particular, it is NP-complete to determine the value of the grand coalition.

Theorem

Let (N,v) a TU game specified with the synergy representation and the value of the grand coalition.
Then we can determine in polynomial time whether the core of the game is empty.

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

Multi-issue representation

Some coalitions may form to solve problems requiring distinct competences. For example, solving a set of tasks requiring different expertises.

Definition (Decomposition)

The vector of characteristic functions $\langle v_1, v_2, \ldots, v_T \rangle$, with each $v_i : 2^N \to \mathbb{R}$, is a **decomposition** over T issues of characteristic function $v : 2^N \to \mathbb{R}$
if for any $S \subseteq N$, $v(S) = \sum_{i=1}^{T} v_i(S)$.

It is a fully expressive representation (can use 1 issue).

Multi-issue representation

Theorem

The Shapley value of a coalitional game represented with multi-issue representation can be computed in linear time.

Theorem

Checking whether a given value division is in the core is coNP-complete.

V. Conitzer and T. Sandholm. **Computing shapley values, manipulating value division schemes, and checking core membership in multi-issue domains.** In *Proc. of the 19th Nat. Conf. on Artificial Intelligence (AAAI-04)*

A logical approach: Marginal contribution nets (MC-nets)

The idea:

- represent each player by a boolean variable,
- treat the characteristic vector of a coalition as a truth assignment.
- the truth assignment can be used to check whether a formula is satisfied and to compute the value of a coalition.

Let N be a collection of atomic variables.

Definition (Rule)

A **rule** has a syntactic form (ϕ, w) where ϕ is called the pattern and is a boolean formula containing variables in N and w is called the weight, and is a real number.

examples:
$(a \wedge b, 5)$: each coalition containing both agents a and b increase its value by 5 units.
$(b,2)$: each coalition containing b increase its value by 2.

A logical approach: Marginal contribution nets (MC-nets)

Definition (Marginal contribution nets)

An MC-net consists of a set of rules $\{(p_1,w_1),\ldots(pk,w_k)\}$ where the valuation function is given by

$$v(\mathcal{C}) = \sum_{i=1}^{k} p_i(e^{\mathcal{C}}) w_i,$$

where $p_i(e^{\mathcal{C}})$ evaluates to 1 if the boolean formula p_i evaluates to true for the truth assignment $e^{\mathcal{C}}$ and 0 otherwise.

S. Ieong and Y. Shoham, **Marginal contribution nets: a compact representation scheme for coalitional games**, in *Proceedings of the 6th ACM conference on Electronic commerce*, 2005.

Examples

Let us consider an MC-net with the following two rules:

$$(a \wedge b, 5) \text{ and } (b,2)$$

The coalitional game represented has two agents a and b and the valuation function is defined as follows:

$$\begin{array}{ll} v(\emptyset) = 0 & v(\{b\}) = 2 \\ v(\{a\}) = 0 & v(\{a,b\}) = 5+2 = 7 \end{array}$$

We can use negations in rules, and negative weights. Let consider the following example:

$$(a \wedge b, 5),\ (b,2),\ (c,4) \text{ and } (b \wedge \neg c, -2)$$

$$\begin{array}{lll} v(\emptyset) = 0 & v(\{b\}) = 2-2 = 0 & v(\{a,c\}) = 4 \\ v(\{a\}) = 0 & v(\{a,b\}) = 5+2-2 = 5 & v(\{b,c\}) = 4+2 = 6 \end{array}$$

Theorem (Expressivity)

- MC-nets can represent **any game** when negative literals are allowed in the patterns, or when the weights can be negative.
- When the patterns are limited to conjunctive formula over postivie literals and weights are nonnegative, MC-nets can represent all and only **convex games**.

Proposition

MC-nets generalize the induced subgraph game representation (strict generalization) and the multi-issue representation.

Theorem

Given a TU game represented by an MC-net limited to conjunctive patterns, the **Shapley value** can be computed in time **linear** in the size of the input.

Proof sketch: we can treat each rule as a game, compute the Shapley value for the rule, and use ADD to sum all the values for the overall game. For a rule, we cannot distinguish the contribution of each agent, by SYM, they must have the same value. It is a bit more complicated when negation occurs (see Ieong and Shoham, 2005).

Proposition

Determining whether the core is empty or checking whether an imputation lies in the core are coNP-hard.

Proof sketch: due to the fact that MC-nets generalize over weighted graph games.

Summary

- We saw some representations that are more succinct or that may help to compute faster some solution concepts.
- representation for specific games (not complete): induced subgraph game, superadditive representation
- general representations, that may require less space in some cases (multi-issue, MC-nets)
- Computing some solution concepts become easy in some case (Shapley value with WGG, Multi-issue, MC nets; empty core for superadditive representation).

Coming next

- NTU games