CZECH TECHNICAL UNIVERSITY IN PRAGUE

0I OTEVŘENÁ INFORMATIKA

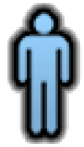
Cooperative Game Theory

Michal Jakob
Agent Technology Center,
Dept. of Computer Science and Engineering,
FEE, Czech Technical University

AE4M36MAS Autumn 2013 - Lecture 8

Where are We?

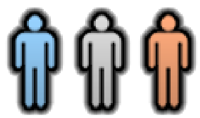
Agent architectures (inc. BDI architecture)



Logics for MAS

Non-cooperative game theory

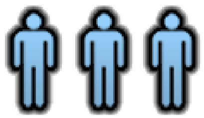
Cooperative game theory



Auctions

Social choice

Distributed constraint reasoning
(satisfaction and optimization)



Motivating Example: Car Pooling

People drive to work and would like to form car pools.

- Some can pick up others on their way to work. Others have to go out of their way to pick up others.
- A car can only hold 5 people.

Assume people care about (1) **money** and (2) **time** and it is possible to convert between the two.

Who *should carpool* ***together****?* ***How much*** *should they* ***pay*** *each other? How often should they each drive?*

Concerns

Rationality

- by carpooling I should save more money than I loose time

Fairness

- saving in money and loses in time should be fairly distributed

Cooperative game theory formalizes such notions

Outline

1. Introduction
2. Basic definitions
3. Solution concepts
4. Compact representations
5. Coalition structure generation
6. Conclusion

Cooperative Game Theory

Model of **coalition (team) formation**

- friends agreeing on a trip
- entrepreneurs trying to form small companies
- companies cooperating to handle a large contract

Assumes a **coalition** can **achieve more** than (the sum of) individual agents

- Better to team up and split the payoff than receive payoff individually

Also called **coalitional game theory**

Called cooperative but **agents still pursue their own interests**

Non Cooperative vs. Cooperative GT

Non-cooperative GT	**Cooperative GT**
Payoffs go **directly to individual** agents	Payoffs go **to coalitions** which **redistribute** them to their members*
Players choose an **action**	Players choose a **coalition** to join and agree on **payoff distribution**
Model of **strategic confrontation**	Model of **team / cooperation formation**
Players are **self-interested**	

*transferable utility games

Example: Task Allocation

A **set of tasks** needs to be performed

- they require different expertises
- they may be decomposed.

Agents do not have enough resource on their own to perform a task.

Find **complementary agents** to perform the tasks

- robots have the ability to move objects in a plant, but multiple robots are required to move a heavy box.
- transportation domain: agents are trucks, trains, airplanes, ships... a task is a good to be transported.

Example: Voting Game

The parliament of Micronesia is made up of **four political parties**, A, B, C, and D, which have **45, 25, 15, and 15 representatives**, respectively.

They are to vote on whether to pass a $100 million spending **bill** and how much of this amount should be controlled by each of the parties.

A **majority vote**, that is, a **minimum of 51 votes**, is required in order to pass any legislation, and if the bill does not pass then every party gets zero to spend.

Example: Buying Ice-cream

$\boldsymbol{n}$ **children**, each has some amount of money

- the i-th child has b_i dollars

Three types of ice-cream tubs are for sale:

- Type 1 costs \$7, contains 500g
- Type 2 costs \$9, contains 750g
- Type 3 costs \$11, contains 1kg

Children have **utility for ice-cream**, and do not care about **money**

The **payoff of each group**: the maximum quantity of ice-cream the members of the group can buy by pooling their money

The ice-cream can be **shared arbitrarily** within the group

How Is a Cooperative Game Played?

1. Knowing the payoffs for different coalitions, agents analyze which coalitions and which payoff distributions would be beneficial for them.
2. Agents agree on coalitions and payoff distributions
   - requires contracts – infrastructure for cooperation
3. Task is executed and gain distributed.

We will now see how to formalize these ideas.

Basic Definitions

Cooperative Game Theory

Coalitional Games

TRANSFERABLE UTILITY GAMES

Payoffs are given **to the group** and then divided among its members.

Satisfied whenever there is a **universal currency** that is used for exchange in the system.

NON-TRANSFERABLE UTILITY GAMES

Group actions result in **payoffs to individual** group members.

Coalitional Game

Transferable utility assumption: the payoffs to a coalition may be freely redistributed among its members.

Definition (Coalitional game with transferable utility)

A **coalitional game with transferable utility** is a pair $(\boldsymbol{N}, \boldsymbol{v})$ where

- N is a finite set of players (also termed **grand coalition**), indexed by i; and
- $v: 2^N \mapsto \mathbb{R}$ is a **characteristic function (**also termed **valuation function**) that associates with each coalition $S \subseteq N$ a real-valued **payoff $\boldsymbol{v}(\boldsymbol{S})$** that the coalition's members can distribute among themselves. We assume $v(\emptyset) = 0$.

Simple Example

$$N = \{1,2,3\}$$

S	$v(S)$
(1)	2
(2)	2
(3)	4
(12)	5
(13)	7
(23)	8
(123)	9

Illustrative Example

C: \$4  M: \$3  P: \$3 

 w = 500 p = \$7

 w = 750 p = \$9

w = 1000 p = \$11

Characteristic function $v(C)$

- $v(\emptyset) = v(\{C\}) = v(\{M\}) = v(\{P\}) = 0$
- $v(\{C,M\}) = 500, v(\{C,P\}) = 500, v(\{M,P\}) = 0$
- $v(\{C,M,P\}) = 750$

Superadditive Games

Definition (Superadditive game)

A **coalitional game** (N, v) is called **superadditive** if $v(C \cap D) \geq v(C) + v(D)$ for every pair of disjoint coalitions $C, D \subseteq N$.

In superadditive games, two coalitions can always **merge** without losing money (i.e. their members can work without interference); hence, we can assume that players form the **grand coalition.**

Outcome and Payoff Vector

Definition (Outcome and Payoff)

An **outcome** of a game (N, v) is a pair $(CS, \vec{x})$ where

- $CS = (C_1, \dots, C_k)$, $\bigcup_i C_i = N$, $C_i \cap C_j = \emptyset$ for $i \neq j$, is a **coalition structure**, i.e., a partition of N into coalitions.
- $\vec{x} = (x_1, \dots, x_n)$, $x_i \geq 0$ for all $i \in N$, $\sum_{i \in C} x_i = v(C)$ for each $C \in CS$, is a **payoff (distribution) vector** which distributes the value of each coalition in CS to the coalition's members.

Payoff is **individually rational** if $x_i \geq v(\{a_i\})$

Note: Coalition structure often not explicitly mentioned

- grand coalition assumed in the case of superadditive games

Example

S	$v(S)$
(1)	2
(2)	2
(3)	4
(12)	5
(13)	7
(23)	8
(123)	9

Outcome examples

(1)(2)(3)
$2 + 2 + 4 = 8$

(1)(23)
$2 + 8 = 10$

$\vec{x} = (2, 4, 4)$
$\vec{x} = (3, 4, 3)$

not individually rational

(2)(13)
$2 + 7 = 9$

(123)
9

(3)(12)
$4 + 5 = 9$

$\vec{x} = (2, 3, 4)$

not stable

Solution Concepts

Cooperative Games

Solution Concepts

What are the outcomes that are likely to arise in cooperative games?

Rewards from cooperation need to be divided in a motivating way.

Two concerns:

1. **Stability**: What the incentives are for agents to stay in a coalition structure?
2. **Fairness**: How well payoffs reflect each agent's contribution?

What Is a Good Outcome?

C: \$4  M: \$3  P: \$3 

Characteristic function

- $v(\emptyset) = v(\{C\}) = v(\{M\}) = v(\{P\}) = 0$
- $v(\{C, M\}) = 500, v(\{C, P\}) = 500, v(\{M, P\}) = 0$
- $v(\{C, M, P\}) = 750$

How should the players share the ice-cream?

- if they share as (200, 200, 350), Charlie and Marcie can get more ice-cream by buying a 500g tub on their own, and splitting it equally
- the outcome (200, 200, 350) is **not stable**!

The Core

Under **what payment** distributions is the outcome of a game stable?

- As long as each coalition earns at least as much as it can make on its own.
- This is the case if and only if the payoff vector is drawn from a set called the core.

Definition (Core)

A payoff vector $\vec{x}$ is in the **core** of a coalitional game (N, v) iff

$$\forall S \subseteq N, \sum_{i \in S} x_i \geq v(S)$$

The **core** of a game is the set of **all stable outcomes**, i.e., outcomes that no coalition wants to deviate from.

- analogue to strong Nash equilibrium (allows deviations by groups of agent)

Ice-Cream Game: Core

C: \$4  M: \$3  P: \$3 

$v(\emptyset) = v(\{C\}) = v(\{M\}) = v(\{P\}) = 0, v(\{C, M\}) = 500, v(\{C, P\}) = 500, v(\{M, P\}) = 0, v(\{C, M, P\}) = 750$

(200, 200, 350) **is not** in the core:

- $v(\{C, M\}) > x_C + x_M$

(250, 250, 250) **is** in the core:

- no subgroup of players can deviate so that each member of the subgroup gets more

(750, 0, 0) **is** also in the core:

- Marcie and Pattie cannot get more on their own! → *fairness*?

Core: Example

S	$v(S)$
(1)	1
(2)	2
(3)	2
(12)	4
(13)	3
(23)	4
(123)	6

$\sum_{i \in S} x_i$	$\sum_{i \in S} x'_i$	$\sum_{i \in S} x''_i$
2	2	1
1	2	3
2	2	2
3	4	3
4	4	3
3	4	5
5	6	6

? In the core, i.e., $\forall S \subseteq N, \sum_{i \in S} x_i \geq v(S)$?

$\vec{x} = (2, 1, 2)$ No

$\vec{x}' = (2, 2, 2)$ Yes

$\vec{x}'' = (1, 2, 3)$ No

Core: Existence

S	$v(S)$
(1)	0
(2)	0
(3)	0
(12)	10
(13)	10
(23)	10
(123)	10

? Is the core always non-empty?

No. Core existence guaranteed only for certain special subclasses of games.
Core is also **not unique** (there might be infinitely many payoff divisions in the core).

ε-Core

If the core is empty, we may want to find approximately stable outcomes

Need to relax the notion of the core:

- core: $x(C) \geq v(C)$ for all $C \subseteq N$
- ε-**core**: $x(C) \geq v(C) - \varepsilon$ for all $C \subseteq N$

<u>Example:</u>

$N = \{1, 2, 3\}$, $v(C) = 1$ if $|C| > 1$, $v(C) = 0$ otherwise

- 1/3-core is non-empty: (1/3, 1/3, 1/3)$\in$ 1/3-core
- ε-core is empty for any ε < 1/3:

  <= $x_i \geq 1/3$ for some $i = 1, 2, 3$, so $x(N \setminus \{i\}) \leq 2/3$, $v(N\{i\}) = 1$

Least Core

If an outcome $\vec{x}$ is in ε-core, the deficit $v(C) - \vec{x}(C)$ of any coalition is at most ε.

We are interested in outcomes that **minimize** the **worst-case deficit**.

Let $\varepsilon^*(G) = \inf\{\varepsilon | \varepsilon\text{−core of } G \text{ is not empty}\}$

- it can be shown that $\varepsilon^*(G)$-core is not empty

Definition: $\varepsilon^*(G)$-core is called the **least core** of G

- $\varepsilon^*(G)$ is called the value of the least core

Example (previous slide): least core = 1/3-core

Further Solution Concepts

Nucleolus

Bargaining set

Kernel

more complicated
stability considerations

Distributing Payments

How should we *fairly* distribute a coalition's payoff?

S	$v(S)$
()	0
(1)	1
(2)	3
(12)	6

? If the agents form (12), how much should each get paid?

Fairness: Axiomatic Approach

What is fair?

Axiomatic approach – a fair payoff distribution should satisfy:

- **Symmetry:** if two agents *contribute the same*, they should receive the same pay-off (they are interchangeable)
- **Dummy player:** agents that *do not add value* to any coalition should get what they earn on their own
- **Additivity:** if two *games are combined*, the value a player gets should be the sum of the values it gets in individual games

Axiomatizing Fairness: Symmetry

Agents i and j **are interchangeable** if they always contribute the same amount to every coalition of the other agents.

- for all S that contains neither i nor j, $v(S \cup \{i\}) = v(S \cup \{j\})$.

The symmetry axiom states that such agents should receive the same payments.

Axiom (Symmetry)

If i and j are interchangeable, then $x_i = x_j$.

Axiomatizing Fairness: Dummy Player

Agent i is a **dummy player** if the amount that i contributes to any coalition is exactly the amount that i is able to achieve alone.

- for all S such that $i \notin S$: $v(S \cup \{i\}) - v(S) = v(\{i\})$.

The dummy player axiom states that dummy players should receive a payment equal to exactly the amount that they achieve on their own.

Axiom (Dummy player)
If i is a dummy player, then $x_i = v(\{i\})$.

Axiomatizing fairness: Additivity

Consider two different coalitional game theory problems, defined by two different characteristic functions v' and v'', involving the same set of agents.

The **additivity axiom** states that if we re-model the setting as a single game in which each coalition S achieves a payoff of $v'(S) + v''(S)$, the agents' payments in each coalition should be *the sum of the payments* they would have achieved for that coalition under *the two separate games*.

Axiom (Additivity)

If $\vec{x}'$ and $\vec{x}''$ are payment distributions in the game (N, v') and (N, v''), respectively, then $x_i^+ = x_i' + x_i''$ where $\vec{x}^+$is the payment distribution in a game $(N, v' + v'')$.

Shapley Value

Theorem

Given a coalitional game (N, v), there is a **unique payoff division** $\vec{\phi}(N, v)$ that divides the full payoff of the grand coalition and that satisfies the Symmetry, Dummy player and Additivity axioms.

This payoff division is called **Shapley value**

Lloyd F. Shapley. 1923–. Responsible for the core and Shapley value solution concepts.

Shapley Value

Definition (Shapley value)

Given a coalitional game (N, v), the **Shapley value** of player i is given by

$$\phi_i(N, v) = \frac{1}{N!} \sum_{S \subseteq N \setminus \{i\}} |S|!\,(|N| - |S| - 1)!\,[v(S \cup \{i\}) - v(S)]$$

This captures the "**average marginal contribution**" of agent i, **averaging** over all the **different sequences** according to which the grand coalition could be built up from the empty coalition.

Shapley Value: Example

? If they form (12), how much should each get paid?

S	$v(S)$
()	0
(1)	1
(2)	3
(12)	6

$$\phi_1 = \frac{1}{2}\big(v(1) - v(\,) + v(21) - v(2)\big)$$
$$= \frac{1}{2}(1 - 0 + 6 - 3) = 2$$

$$\phi_2 = \frac{1}{2}\big(v(2) - v(\,) + v(12) - v(1)\big)$$
$$= \frac{1}{2}(3 - 0 + 6 - 1) = 4$$

Shapley Value: Ice Cream Example

C: \$4  M: \$3  P: \$3 

$v(\emptyset) = v(\{C\}) = v(\{M\}) = v(\{P\}) = 0, v(\{C, M\}) = 500, v(\{C, P\}) = 500, v(\{M, P\}) = 0, v(\{C, M, P\}) = 750$

?

$\phi_C = \frac{1}{3!}\Big(v(C) - v(\,) + v(CM) - v(M) + v(CP) - v(P) +$

Classes of Coalition Games

Superadditive game

Additive game

Constant-sum game

Convex game

Simple game

Convex Games

An important subclass of superadditive games

Definition (Convex game)

A **coalitional game** (N, v) is termed **convex** if $v(C \cup D) \geq v(C) + v(D) - v(C \cap D)$ for every pair of coalitions $C, D \subseteq N$.

Convexity is a **stronger condition** than superadditivity.

- "a player is more useful when he joins a bigger coalition"

Convex games have a number of useful properties

- the **core** is always **non-empty**
- **Shapley value** is **in the core**

Simple Games

Definition (Simple game)

A **coalitional game** (N, v) is termed **simple** if $v(C) \in \{0,1\}$ for any $C \subseteq N$ and v is **monotone**, i.e., if $v(C) = 1$ and $C \subseteq D$, then $v(D) = 1$.

Model of yes/no voting systems.

A coalition C in a simple game is said to be **winning** if $v(C) = 1$ and **losing** if $v(C) = 0$.

A player i in a simple game is a **veto player** if $v(C) = 0$ for any $C \subseteq N \setminus \{i\}$

- equivalently, by monotonicity, v(N\{i}) = 0.

Traditionally, in simple games an outcome is identified with a payoff vector for N.

<u>**Theorem**</u>: A simple game has a **non-empty core** iff it has a **veto player**.

Relation of Game Clases

$$
\begin{array}{rl}
\text{Superadditive} \supset \text{Convex} & \\
 & \text{Additive} \\
\text{Constant sum} & \\
 & \text{Proper simple} \\
\text{Simple} &
\end{array}
$$

Representation Aspects

Compact Representations

A naive representation of a coalition game is infeasible (**exponential** in the number of agents):

- e.g. for three agents $\{1, 2, 3\}$:

$(1) = 5$

$(2) = 5$

$(3) = 5$

$(1, 2) = 10$

$(1, 3) = 10$

$(2, 3) = 20$

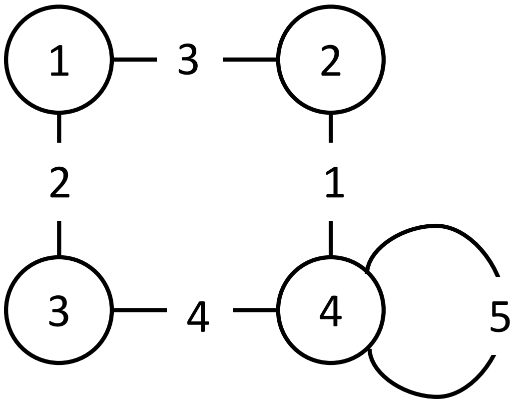
$(1, 2, 3) = 25$

We need a **succinct** representations.

Complete (can represent any game but not necessarily succinct) vs. **incomplete** (represent an important subclass).

Induced Subgraph (Weighted Graph) Games

Define a characteristic function by an undirected weighted graph. Value of a coalition $S \subseteq N$: $v(S) = \sum_{\{i,j\} \subseteq S} w_{i,j}$



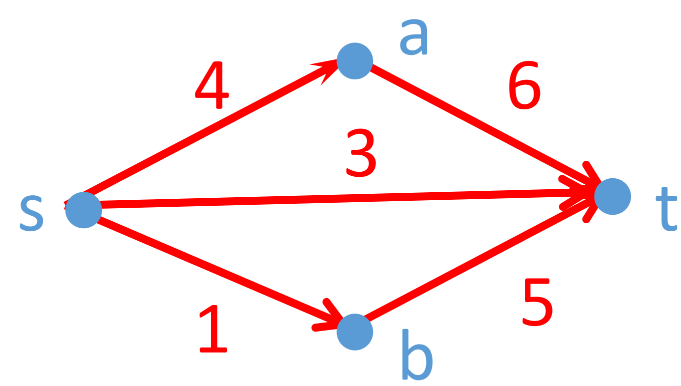
$v(\{1,2,3\}) = 3 + 2 = 5$
$v(\{4\}) = 5$
$v(\{2,4\}) = 1 + 5 = 6$
$v(\{1,3\}) = 2$

If all edge weights are non-negative, this game is convex => non-empty core.

Not a complete representation (not all characteristic functions can be represented)

But **easy to compute** the Shapley value for a given agent in polynomial time: $sh_i = \frac{1}{2}\sum_{j \neq i} w_{i,j}$

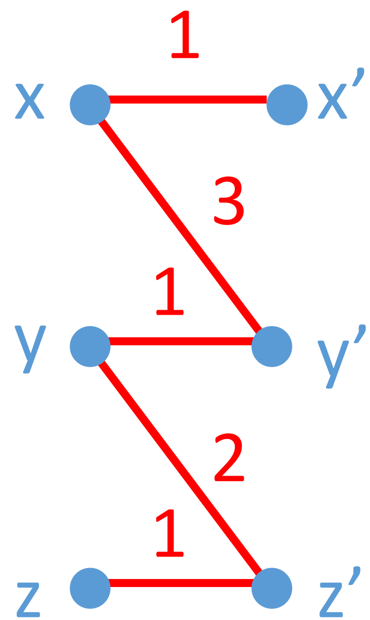
Other Combinatorial Representations



Network flow games

- agents are edges in a network with source s and sink t
- value of a coalition = amount of s–t flow it can carry

Assignment games



- Players are vertices of a bipartite graph
- Value of a coalition = weight of the max-weight induced matching

Matching games

- generalization of assignment games to other than bipartite graphs

Weighted Voting Games

- For each agent $i \in Ag$ define a weight w_i and an overall **quota** q
- A coalition is winning if the sum of their weights exceeds the quota:
$$\nu(C) = \begin{cases} 1 & \text{if } \sum_{i \in C} w_i \geq q \\ 0 & \text{otherwise} \end{cases}$$
- Example: **Simple majority voting**, $w_i = 1$ and $q = \frac{\lceil |Ag|+1 \rceil}{2}$
- Succinct (but incomplete) representation: $\langle q; w_1, \ldots, w_n \rangle$
- Extension: ***k*-weighted voting games** are a complete representation
  - overall game = "conjunction" k of k different weighted voting games
  - Winning coalition is the one that wins in all component games
  - **Game dimension**: k is at most exponential in the number of players
  - Checking whether a k-weighted voting game is minimal is NP-complete

Weighted Voting Games: UK

United Kingdom, 2010:

- 650 seats, $q = 326$
- Conservatives (C): 307
- Labour (L): 258
- Liberal Democrats (LD): 57
- 8 other parties (O), with a total of 28 seats

$N = \{C, L, LD, O\}$

$v(\{C, L\}) = v(\{C, LD\}) = v(\{C, O\}) = 1$

$v(\{L, LD\}) = v(\{L, O\}) = v(\{LD, O\}) = 0$, $v(\{L, LD, O\}) = 1$

L, LD and O are symmetric

$\phi_C = 1/2$, $\phi_L = \phi_{LD} = \phi_O = 1/6$

Shapley Value in Weighted Voting Games

In a simple game $G = (N, v)$, a player i is said to be pivotal

- for a coalition $C \subseteq N$ if $v(C) = 0$, $v(C \cup \{i\}) = 1$
- for a permutation $\pi \in P(N)$ if he is pivotal for $S_\pi(i)$

In simple games player i's Shapley value =
Pr[i is pivotal for a random permutation]

- measure of voting power

Shapley value is widely used to measure power in various voting bodies

UK elections'10 illustrate that power ≠ weight

Marginal Contribution Nets

- Represent characteristic function as rules: pattern $\longrightarrow$ value
  - the pattern is a conjunction of agents, e.g. $1 \wedge 3$
  - $1 \wedge 3$ would apply to $\{1, 3\}$ and $\{1, 3, 5\}$, but not to $\{1\}$ or $\{8, 12\}$
  - $C \vDash \varphi$, means the rule $\varphi \longrightarrow x$ applies to coalition C
  - $rs_C = \{\varphi \longrightarrow x \in rs | C \vDash \varphi\}$ are the rules that apply to coalition C
- $\nu_{rs}(C) = \sum_{\varphi \longrightarrow x \in rs_C} x$
- Example:
  - $rs_1 = \{a \wedge b \longrightarrow 5,\ b \longrightarrow 2\}$
  - $\nu_{rs_1}(\{a\}) = 0$, $\nu_{rs_1}(\{b\}) = 2$ and $\nu_{rs_1}(\{a, b\}) = 7$
- Extension: allow negation in rules, e.g. $b \wedge \neg c \longrightarrow -2$
- Shapley value can be computed in polynomial time
- Complete representation, but not necessarily succinct

Other Complete Representations

Synergy coalition groups

- only represents values of coalitions of size 1 and those where there is a synergy

Skill-based representation

- agents are assigned a set of skills
- payoff depends on skills in a coalition

Agent-type representation

- agents classified into a small number of types
- characteristic function depends on the number of agents of certain type

Coalition Structure Generation

How do we **partition the set of agents** into coalitions to maximize the overall profit?

Finding Optimal Coalition Structure

Assuming utilitarian solution, i.e., maximizing the total payoff of all coalitions.

Trivial if **superadditive** ➔ **grand coalition**.

Otherwise: **search** for the best coalition **structure**.

The Coalition Structure Generation Problem

Example: given 3 agents, the possible **coalitions** are:

$\{a_1\}$ $\{a_2\}$ $\{a_3\}$ $\{a_1,a_2\}$ $\{a_1,a_3\}$ $\{a_2,a_3\}$ $\{a_1,a_2,a_3\}$

The possible **coalition structures** are:

$\{\{a_1\},\{a_2\},\{a_3\}\}$ $\{\{a_1,a_2\},\{a_3\}\}$ $\{\{a_2\},\{a_1,a_3\}\}$ $\{\{a_1\},\{a_2,a_3\}\}$ $\{\{a_1,a_2,a_3\}\}$

The **input** is the characteristic function

$v(\{a_1\}) = 20$

$v(\{a_2\}) = 40$

$v(\{a_3\}) = 30$

$v(\{a_1,a_2\}) = 70$

$v(\{a_1,a_3\}) = 40$

$v(\{a_2,a_3\}) = 65$

$v(\{a_1,a_2,a_3\}) = 95$

What we want as **output** is a coalition structure in which the **sum of values is maximized**

$V(\ \{\{a_1\},\{a_2\},\{a_3\}\}\ ) = 20+40+30 = 90$

$V(\ \{\{a_1,a_2\},\{a_3\}\}\ ) = 70+30 = \mathbf{100}$

optimal coalition structure

$V(\ \{\{a_2\},\{a_1,a_3\}\}\ ) = 40+40 = 80$

$V(\ \{\{a_1\},\{a_2,a_3\}\}\ ) = 20+65 = 85$

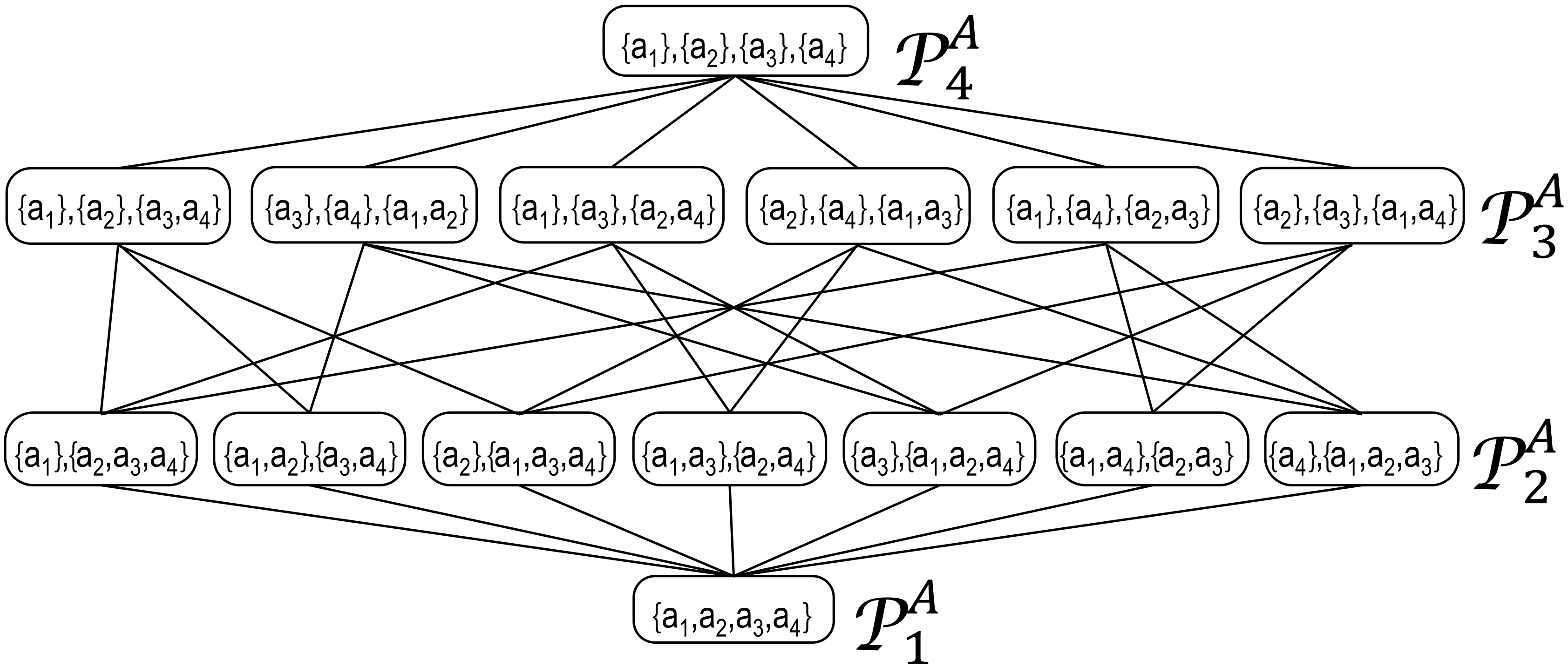
$V(\ \{\{a_1,a_2,a_3\}\} = 95$

Search Space Representation

1. Coalition structure graph
2. Integer partition graph

Coalition Structure Graph (for 4 agents)

$\mathcal{P}_i^A \subseteq \mathcal{P}^A$ contains all coalition structures that consist of exactly i coalitions

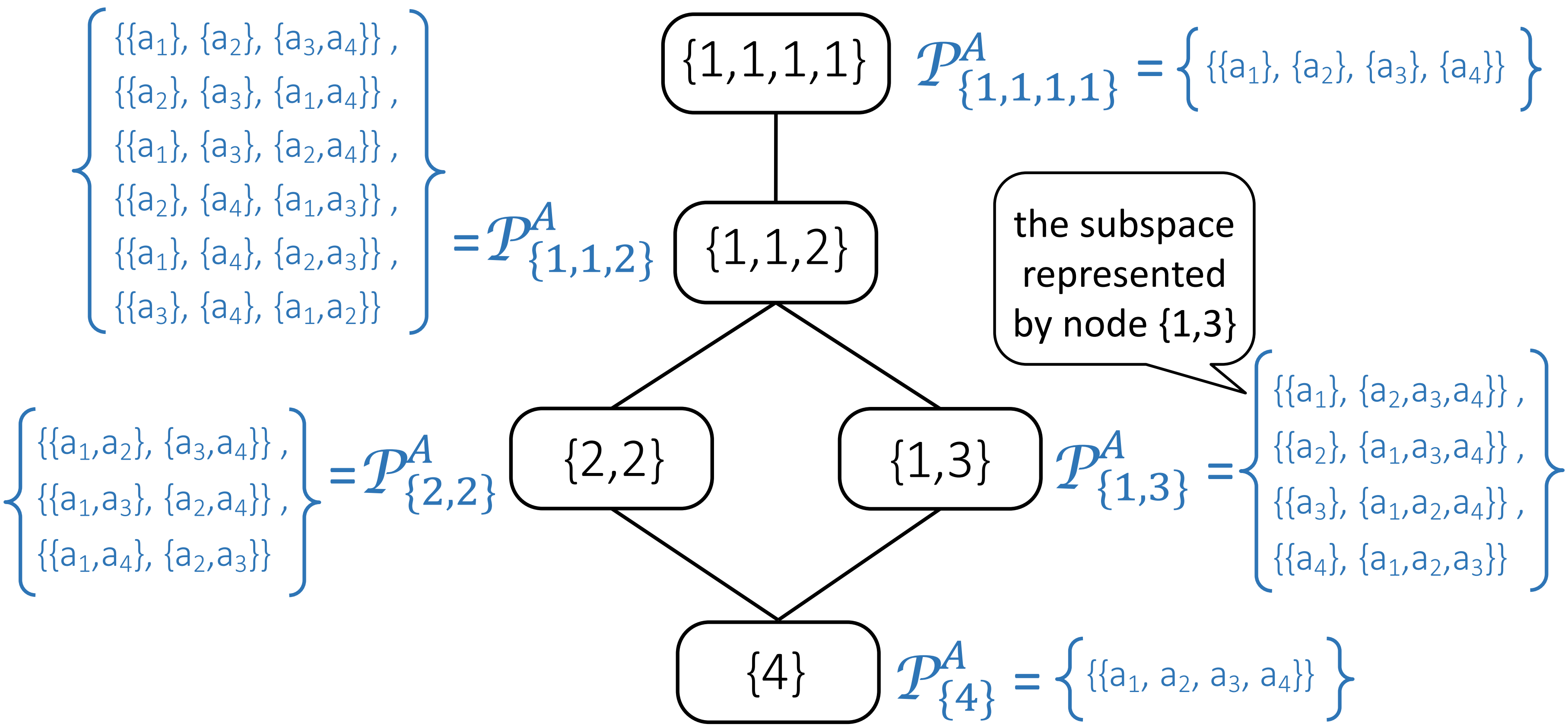


Edge connects two coalition structures iff:

1. they belong to two consecutive levels $\mathcal{P}_i^A$ and $\mathcal{P}_{i-1}^A$
2. the coalition structure in $\mathcal{P}_{i-1}^A$ can be obtained from the one in $\mathcal{P}_i^A$ by merging two coalitions into one

Integer Partition Graph (example of 4 agents)

Every node represents a subspace (coalition sizes match the integers in that node)



Two nodes representing partitions $I, I' \in \mathcal{I}^n$ are connected iff there exists two parts $i, j \in I$ such that $I' = (I \setminus \{i,j\}) \uplus \{i+j\}$

Algorithms for Coalition Formation

Challenge: the number of coalitions for n players:

$$\alpha n^{n/2} \leq B_n \leq n^n$$

for some positive constant α (B_n is a Bell number)

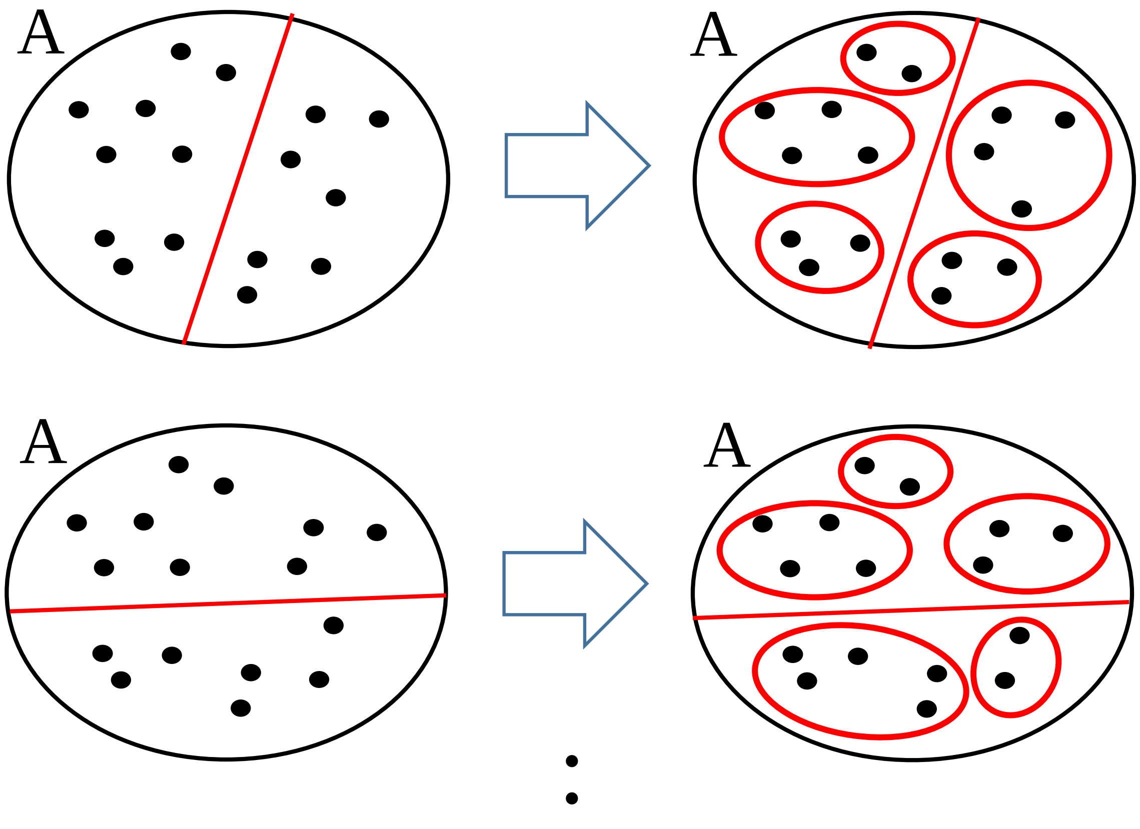
Approaches:

- Optimal: Dynamic programming
- Anytime algorithms with guaranteed bounds
- Heuristics algorithms
- Algorithms for compact representation games

Dynamic Programming (DP) Algorithm

Main observation: To examine all coalition structure CS: $|CS2$, it is sufficient to:

- try the possible ways to split the **set of agents into two sets**, and
- for every half, find the **optimal partition** of that half.

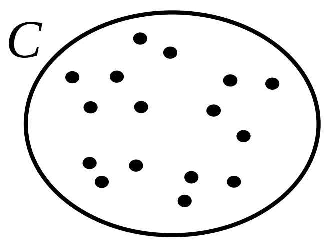


Dynamic Programming (DP) Algorithm

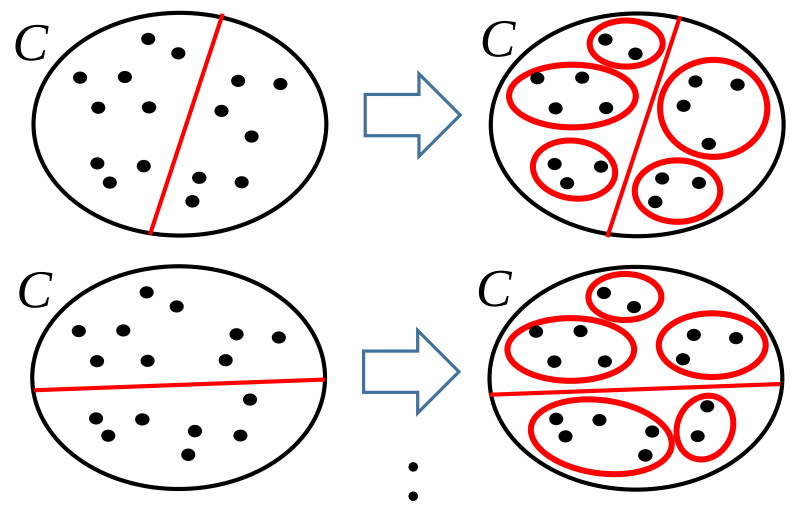
Main theorem: Given a coalition $C \in A$, let $\mathcal{P}^C$ be the set of partitions of C, and let $f(C)$ be the value of an optimal partition of C, i.e., $f(C) = \max_{P \in \mathcal{P}^C} V(P)$. Then,

$$f(C) = \begin{cases} v(C) & \text{if } |C| = 1 \\ \max\left\{v(C), \max_{\{C', C''\} \in \mathcal{P}^C} f(C') + f(C'')\right\} & \text{otherwise} \end{cases}$$

the value of the coalition itself (without partitioning)



the maximum value for all such partitions



Dynamic Programming (DP) Algorithm

Algorithm:

- Iterate over all coalitions $C\colon |C| = 1$, then over all $C\colon |C| = 2$, then all $C\colon |C| = 3$, etc.
- For every coalition, C, compute $f(C)$ using the above equation
- While computing $f(C)$:
  - the algorithm stores in $t(C)$ the best way to split C in two
  - unless it is more beneficial to keep C as it is (i.e., without splitting)
- By the end of this process, $f(A)$ will be computed, which is by definition the value of the optimal coalition structure
- It remains to compute the optimal coalition structure itself, by using $t(A)$

Consider the following example of 4 agents input:

v({1}) = 30
v({2}) = 40
v({3}) = 25
v({4}) = 45
v({1,2}) = 50
v({1,3}) = 60
v({1,4}) = 80
v({2,3}) = 55
v({2,4}) = 70
v({3,4}) = 80
v({1,2,3}) = 90
v({1,2,4}) = 120
v({1,3,4}) = 100
v({2,3,4}) = 115
v({1,2,3,4}) = 140

	coalition	evaluations performed before setting f		t	f
step 1	{1}	v({1})=30		{1}	30
	{2}	v({2})=40		{2}	40
	{3}	v({3})=25		{3}	25
	{4}	v({4})=45		{4}	45
step 2	{1,2}	v({1,2})=50	f({1})+f({2})=70	{1} {2}	70
	{1,3}	v({1,3})=60	f({1})+f({3})=55	{1,3}	60
	{1,4}	v({1,4})=80	f({1})+f({4})=75	{1,4}	80
	{2,3}	v({2,3})=55	f({2})+f({3})=65	{2} {3}	65
	{2,4}	v({2,4})=70	f({2})+f({4})=85	{2} {4}	85
	{3,4}	v({3,4})=80	f({3})+f({4})=70	{3,4}	80
step 3	{1,2,3}	v({1,2,3})=90 f({2})+f({1,3})=100	f({1})+f({2,3})=95 f({3})+f({1,2})=95	{2} {1,3}	100
	{1,2,4}	v({1,2,4})=120 f({2})+f({1,4})=110	f({1})+f({2,4})=115 f({4})+f({1,2})=115	{1,2,4}	120
	{1,3,4}	v({1,3,4})=100 f({3})+f({1,4})=105	f({1})+f({3,4})=110 f({4})+f({1,3})=105	{1} {3,4}	110
	{2,3,4}	v({2,3,4})=115 f({3})+f({2,4})=110	f({2})+f({3,4})=120 f({4})+f({2,3})=110	{2} {3,4}	120
step 4	{1,2,3,4}	v({1,2,3,4})=140 f({2})+f({1,3,4})=150 f({4})+f({1,2,3})=145 f({1,3})+f({2,4})=145	f({1})+f({2,3,4})=150 f({3})+f({1,2,4})=145 f({1,2})+f({3,4})=150 f({1,4})+f({2,3})=145	{1,2} {3,4} step 5	150

Dynamic Programming (DP) Algorithm

Note:

- While DP is guaranteed to find an optimal coalition structure, many of its operations were shown to be redundant
- An improved dynamic programming algorithm (called IDP) was developed that avoids all redundant operations

Advantage:

- IDP is the fastest algorithm that finds an optimal coalition structure in O(3n)

Disadvantage:

- IDP provides no interim solutions before completion, meaning that it is not possible to trade computation time for solution quality.

=> **anytime** algorithms

Anytime Algorithms

An "anytime" algorithm is one whose solution quality improves gradually as computation time increases.

- This way, an interim solution is always available in case the algorithm run to completion.

Advantages:

- agents might not have time to run the algorithm to completion
- being anytime makes the algorithm more robust against failure.

Categories of algorithms

- algorithms based on Identifying Subspaces with Worst-Case Guarantees
- algorithms based on the integer-partition based representation.

Metaheuristic Algorithms

As the number of agents increases, the problem becomes too hard, and the only practical option is to use metaheuristic algorithms.

Advantage:

- Can usually be applied for very large problems.

Disadvantage:

- No guarantees that an optimal solution is ever found
- No guarantees on the quality of their solutions.

Examples:

- Genetic Algorithms [Sen & Dutta, 2000]
- Simulated Annealing [Keinanen, 2009]
- Decentralized greedy algorithm [Shehory & Kraus, 1998]
- Greedy algorithm based on GRASP [Di Mauro et al, 2010]

Conclusions

Cooperative game theory models the formation of **teams of selfish agents.**

- **coalitional game** formalizes the concept
- **core solution** concept address the issue of coalition stability
- **Shapley value** solution concept represents a fair distribution of payments

For practical computation, compact representation of coalition games are required.

For non-superadditive games, coalition structure needs to be found.

Reading:

- [Shoham]: 12.1-12.2
- [Vidal]: Chapter 4
- [Weiss]: Chapter 8