Multi-Agent Systems

University "Politehnica" of Bucarest
Spring 2018

Lecture 9 & 10
Computational Coalition Formation

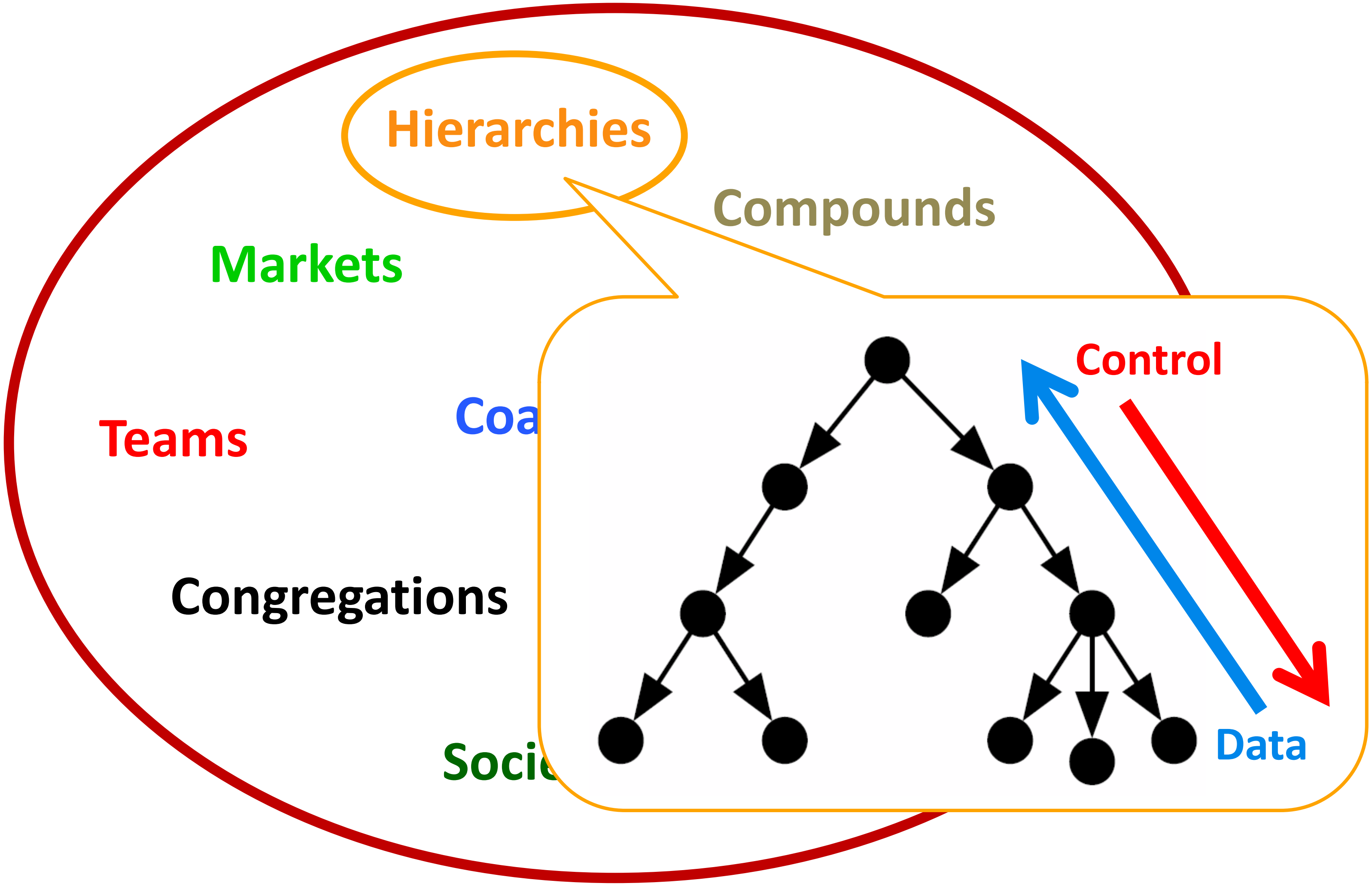
Coalitional Game Theory

how do selfish agents form teams
in order to work together

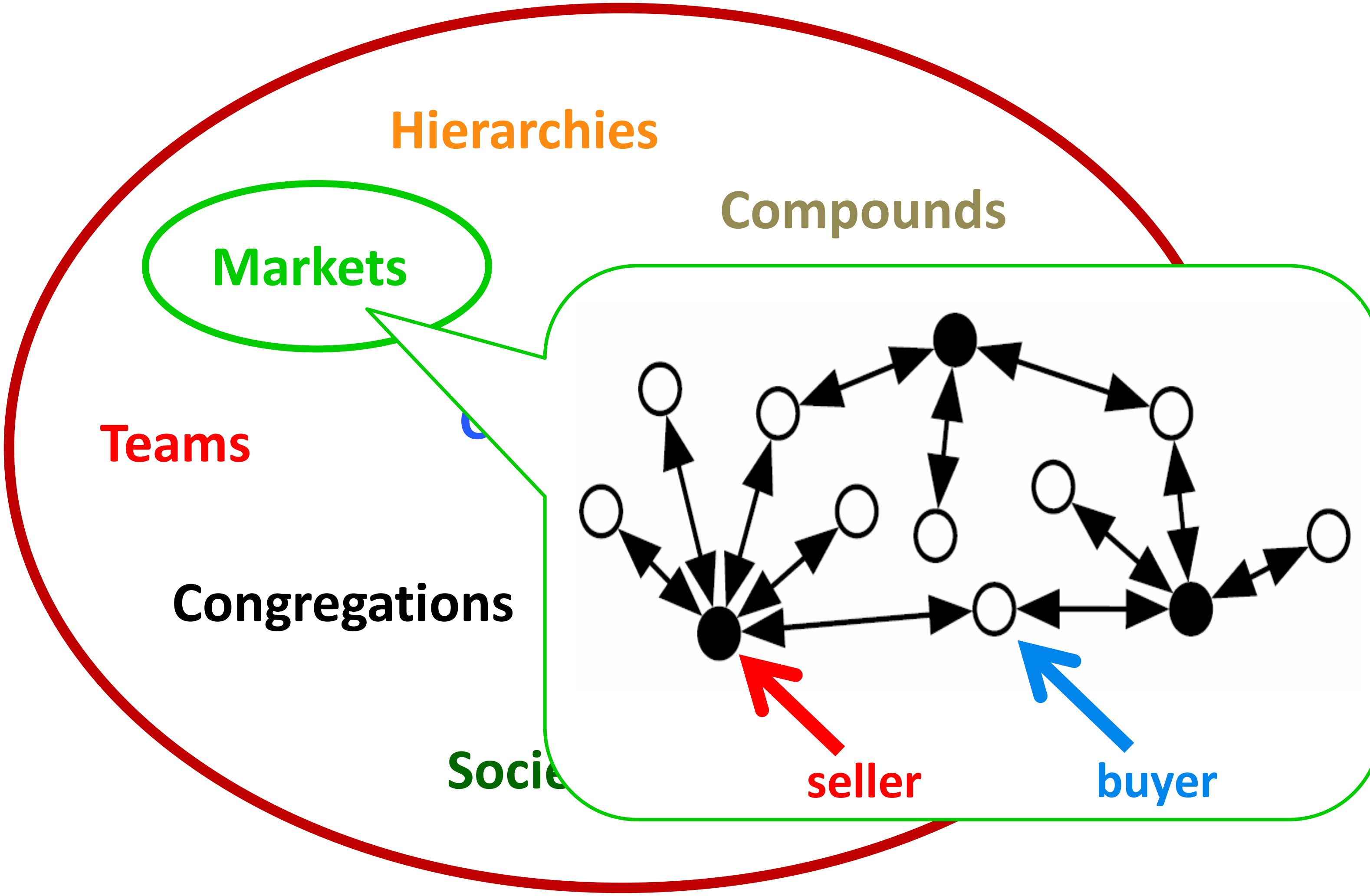
Coalitional Game Theory

- Introduction
- Definitions
- Solution concepts
- Representations
  and computational issues

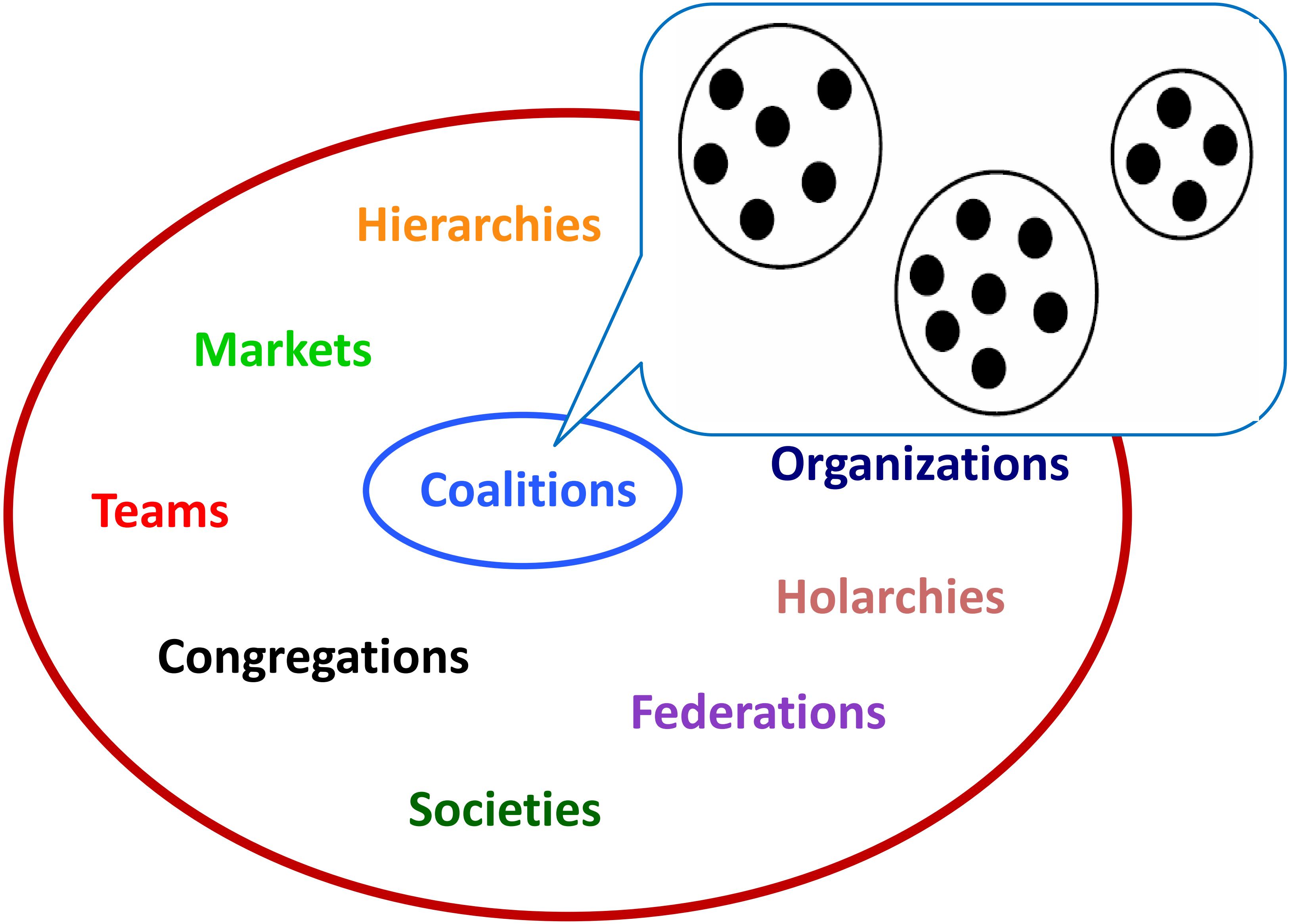
Organizations in Multi-Agent Systems



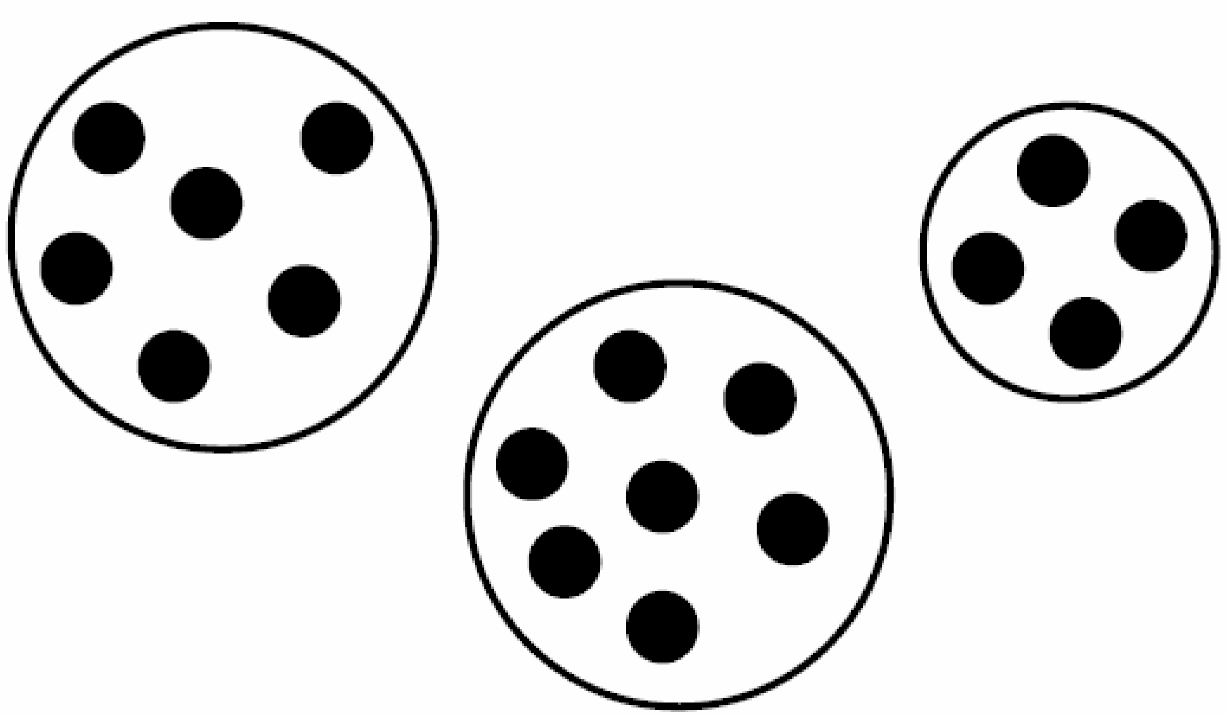
Organizations in Multi-Agent Systems



Organizations in Multi-Agent Systems



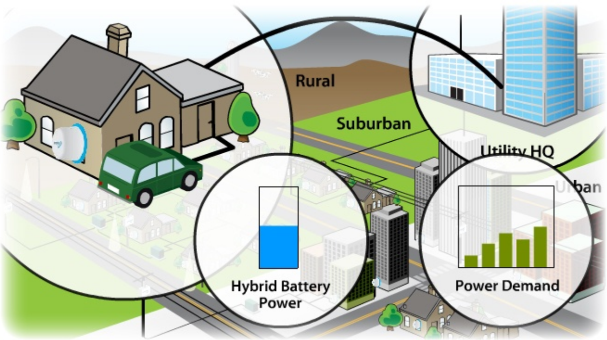
Coalition Formation



Main characteristicis

- Coalitions in general are goal-directed and short-lived
- No coordination among members of different coalitions
- The organizational structure within each coalition is flat

Applications of Coalition Formation



Smart Energy Grids

Intelligent appliances and energy storage devices coordinate for optimal energy use

Electronic-commerce

Cooperation among buyers to obtain quantity discounts, and sellers to maintain cartel pricing.

Disaster Management

UN report said: *"Efforts by the United Nations in Haiti have lacked sufficient coordination“*

Applications of Coalition Formation

- **Distributed sensor networks:** Coalitions of sensors can work together to track targets of interest
- **Distributed vehicle routing:** Coalitions of delivery companies can be formed to reduce the transportation costs by sharing deliveries
- **Information gathering:** Several information servers can form coalitions to answer queries

Cooperative vs. Non-Cooperative Games

- Cooperation does not occur in **non-cooperative games**, because players cannot make binding agreements

But what if binding agreements are possible?

- **Cooperative games** model scenarios, where agents can benefit by cooperating; binding agreements are possible

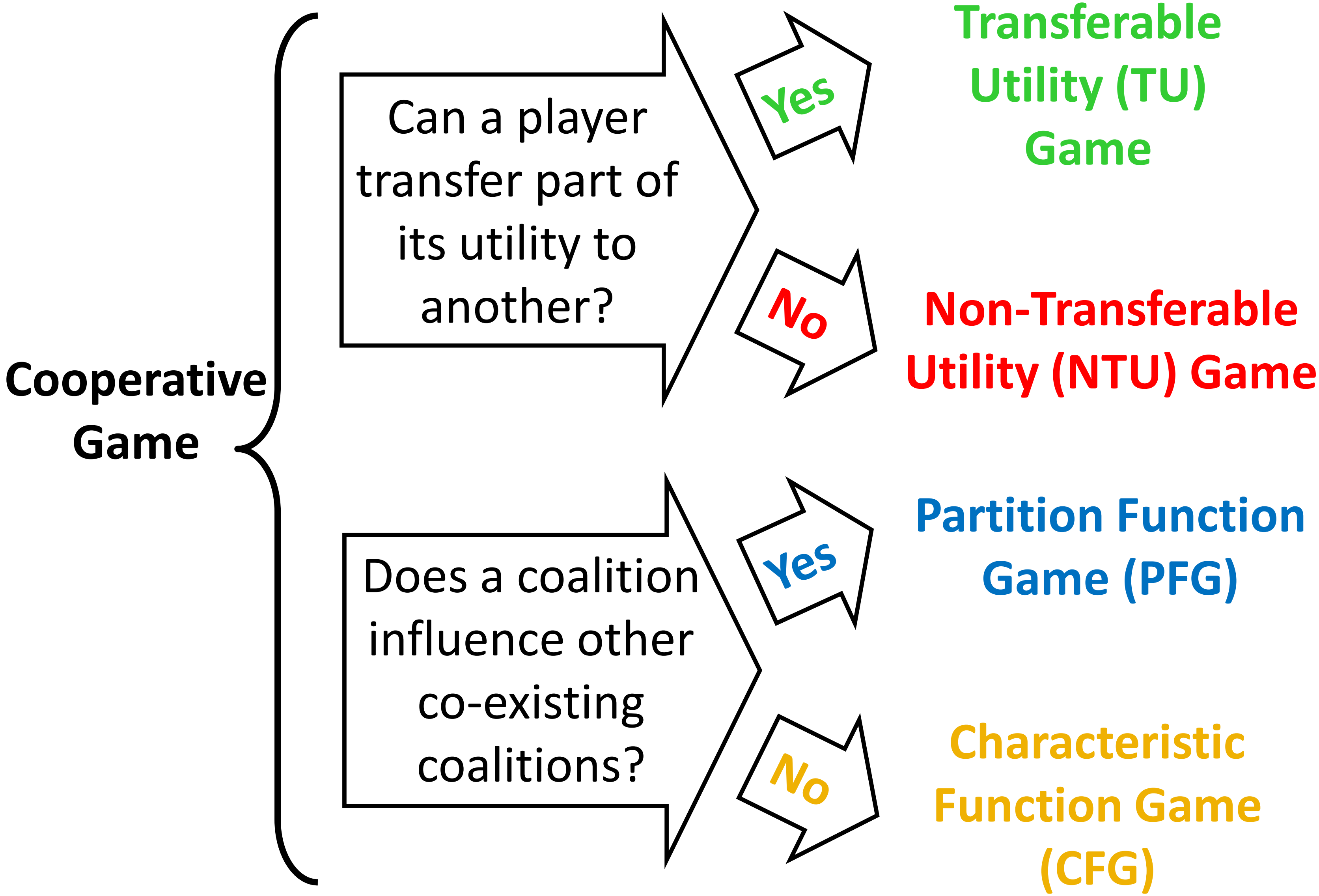
1. Introduction
Cooperative Games

- Cooperative games model scenarios, where
  - agents can benefit by cooperating
  - binding agreements are possible
- In cooperative games, actions are taken by groups of agents

Transferable utility games: payoffs are given to the group and then divided among its members

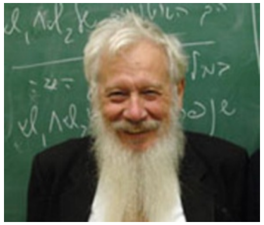
Non-transferable utility games: group actions result in payoffs to individual group members

Cooperative Games



Non-Transferable Utility Games (NTU): Writing Papers

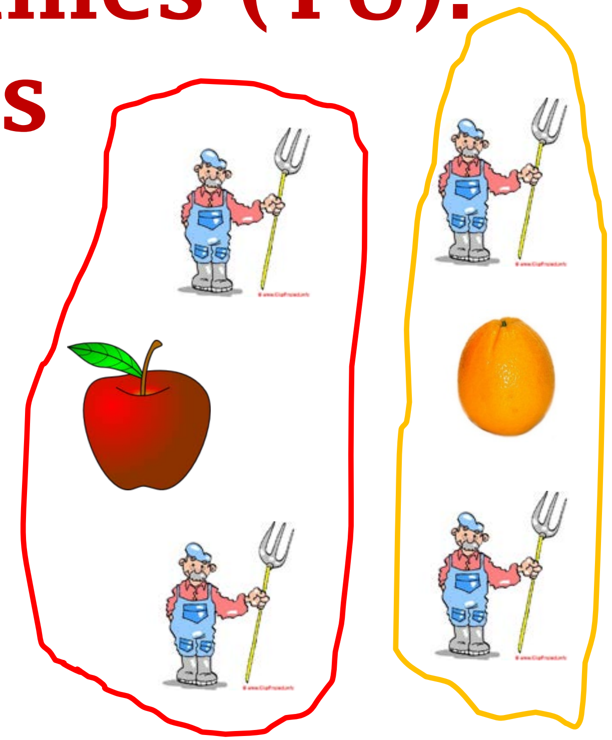
- n researchers working at m different universities can form groups to write papers on game theory
- each group of researchers can work together; the composition of a group determines the quality of the paper they produce
- each author receives a payoff from his own university
  - promotion
  - bonus
- Payoffs are non-transferable






Transferable Utility Games (TU): Happy Farmers

- n farmers can cooperate to grow fruit
- Each group of farmers can grow apples or oranges
- a group of size k can grow f(k) tons of apples and g(k) tons of oranges
- Fruit can be sold in the market
- The profit of each group depends on the quantity and type of fruit it grows, and the market price



Transferable Utility Games (TU): Buying Ice-cream

- 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

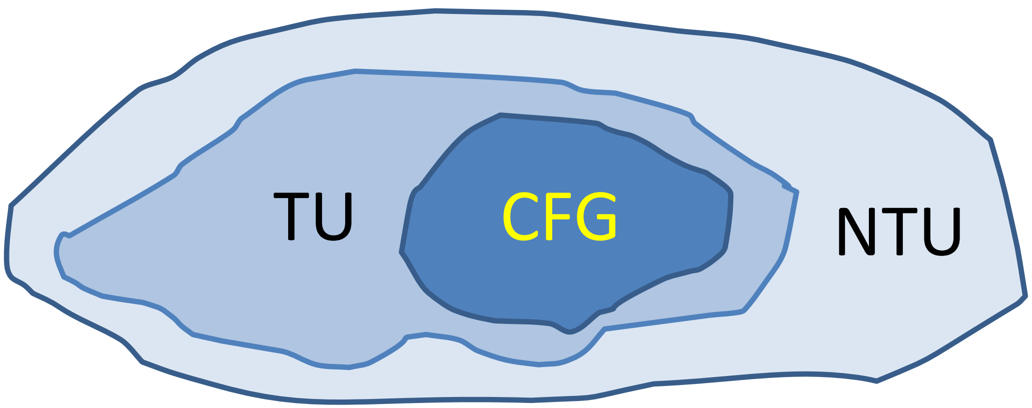
- children have utility for ice-cream, and do not care about money
- The payoff of each group: the maximum quantity of ice-cream the members of the group can buy by pooling their money
- The ice-cream can be shared arbitrarily within the group

Partition Function Games (PFG) vs. Characteristic Function Games (CFG)

- Partition function games (PFG):
- In general TU games, the payoff obtained by a coalition depends on the actions chosen by other coalitions
- Characteristic function games (CFG): the payoff of each coalition only depends on the action of that coalition
  - in such games, each coalition can be identified with the profit it obtains by choosing its best action
  - Happy Farmers game is a PFG, but not a CFG
  - Ice Cream game is a CFG

Classes of Cooperative Games: The Big Picture

- Any TU game can be represented as an NTU game with a continuum of actions
  - each payoff division scheme in the TU game can be interpreted as an action in the NTU game



- **We shall discuss characteristic function games (CFG)**, and use term "TU games" to refer to such games

How Is a Cooperative Game Played?

- Even though agents work together they are still selfish
- The partition into coalitions and payoff distribution should be such that no player (or group of players) has an incentive to deviate
- May also want to ensure that the outcome is fair: the payoff of each agent is proportional to his contribution
- How to formalize these ideas

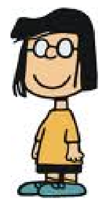
2. Transferable Utility Games (TU) Formalized

- A **transferable utility game** is a pair (N, v), where:
  - $N = \{1, ..., n\}$ is the set of players
  - $v: 2^N \rightarrow \mathbb{R}$ is the **characteristic function**
    - for each subset of players C, $\mathbf{v(C)}$ is the amount that the members of C can earn by working together
  - usually it is assumed that v is
    - normalized: $v(\emptyset) = 0$
    - non-negative: $v(C) \geq 0$ for any $C \subseteq N$
    - monotone: $v(C) \leq v(D)$ for any C, D such that $C \subseteq D$

Value of the coalition

- A **coalition** is any subset of N;
  N itself is called the **grand coalition**

Ice-Cream Game: Characteristic Function


C: \$6,

M: \$4,

P: \$3

w = 500
p = \$7

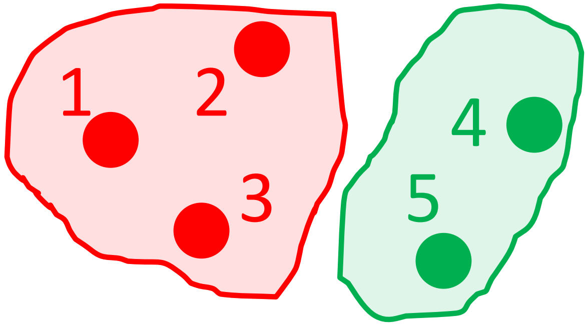
w = 750
p = \$9

w = 1000
p = \$11

- $v(\emptyset) = v(\{C\}) = v(\{M\}) = v(\{P\}) = 0$
- $v(\{C, M\}) = 750$, $v(\{C, P\}) = 750$, $v(\{M, P\}) = 500$
- $v(\{C, M, P\}) = 1000$

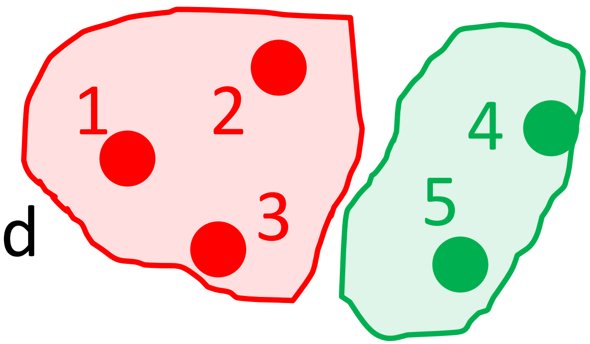
Transferable Utility Games: Outcome

- An outcome of a TU game $G = (N, v)$ is a pair $(CS, \underline{\mathbf{x}})$, where:
  - $CS = (C_1, \ldots, C_k)$ is a coalition structure, i.e., partition of N into coalitions:
    - $\cup_i C_i = N$, $C_i \cap C_j = \emptyset$ for $i \neq j$
  - $\underline{\mathbf{x}} = (x_1, \ldots, x_n)$ is a payoff vector, which distributes the value of each coalition in CS:
    - $x_i \geq 0$ for all $i \in N$
    - $\Sigma_{i \in C}\, x_i = v(C)$ for each C is CS



Transferable Utility Games: Outcome

- Example:
  - suppose v({1, 2, 3}) = 9, v({4, 5}) = 4
  - then (({1, 2, 3}, {4, 5}), (3, 3, 3, 3, 1)) is an outcome
  - (({1, 2, 3}, {4, 5}), (2, 3, 2, 3, 3)) is NOT an outcome: transfers between coalitions are not allowed



- An outcome (CS, **x**) is called an imputation if it satisfies individual rationality: $x_i \geq v(\{i\})$ for all $i \in N$
- Notation: we will denote $\Sigma_{i \in C}\, x_i$ by $x(C)$

Superadditive Games

- Definition: a game $G = (N, v)$ is called **superadditive** if $v(C \cup D) \geq v(C) + v(D)$ for any two disjoint coalitions C and D
- Example: $v(C) = |C|^2$:
  - $v(C \cup D) = (|C|+|D|)^2 \geq |C|^2+|D|^2 = v(C) + v(D)$
- In superadditive games, two coalitions can always merge without losing money; hence, we can assume that players form the grand coalition

Superadditive Games

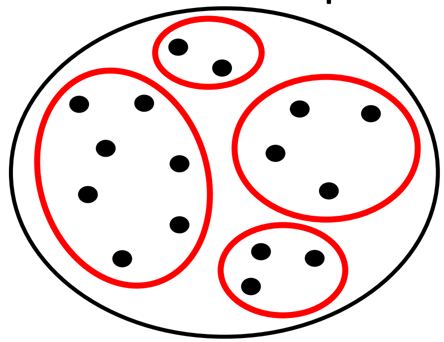
- Convention: in superadditive games, we identify outcomes with payoff vectors for the grand coalition
  - i.e., an outcome is a vector $\underline{\mathbf{x}} = (x_1, \ldots, x_n)$ with $\Sigma_{i \in N} x_i = v(N)$
- Caution: some GT/MAS papers define outcomes in this way even if the game is not superadditive

Superadditive Games

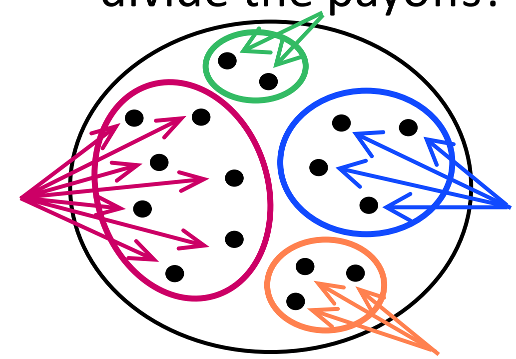
- In super-additive games, the grand coalition forms, so:

Non-Superadditive Game

Which coalition structure is optimal?

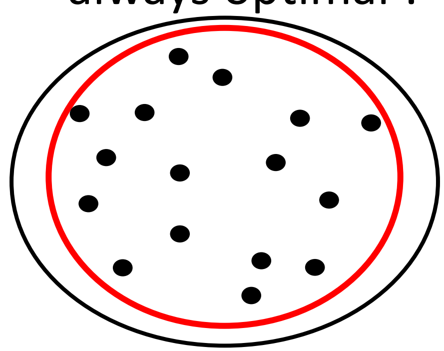


How should we divide the payoffs?

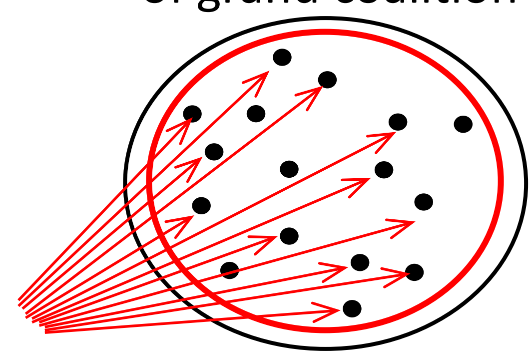


Superadditive Game

Grand coalition is always optimal !



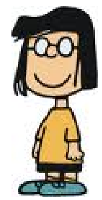
How to divide payoff of grand coalition ?



3. Solution concepts

- **What Is a Good Outcome?**
- Evaluate the outcomes according to two sets of criteria:
- **Stability** = what the incentives are for the agents to stay in the coalition structure.
- **Fairness** = how well each agent's payoff reflects its contribution
- These two sets of criteria give rise to two families of payoff division schemes, or solution concepts.

What Is a Good Outcome?

-  C: $4,  M: $3,  P: $3

w = 500	w = 750	w = 1000
p = $7	p = $9	p = $11

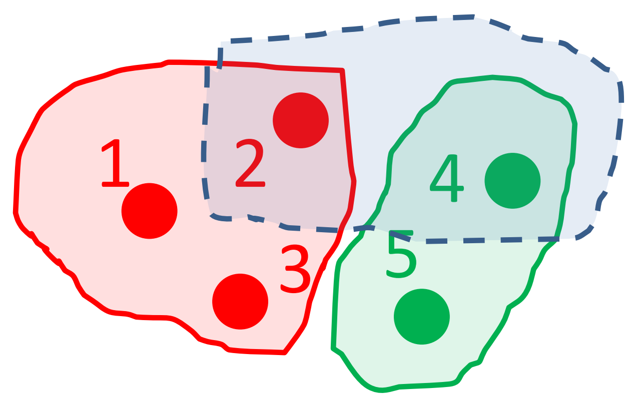
- v(Ø) = v({C}) = v({M}) = v({P}) = 0
- v({C, M}) = 500, v({C, P}) = 500, v({M, P}) = 0
- v({C, M, P}) = 750
- This is a superadditive game
  - outcomes are payoff vectors
- How should the players share the ice-cream?
  - if they share as (200, 200, 350), Charlie and Marcie can get more ice-cream by buying a 500g tub on their own, and splitting it equally
  - the outcome (200, 200, 350) is not stable!

Coalition Stability

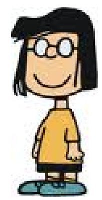
- Definition: the core of a game is the set of all stable outcomes, i.e., outcomes that no coalition wants to deviate from

  core(G) = $\{(CS, \underline{\mathbf{x}}) \mid \Sigma_{i \in C}\, x_i \geq v(C)$ for any $C \subseteq N\}$

  – each coalition earns at least as much as it can make on its own
- Note that G is not assumed to be superadditive
- Example
  – suppose v({1, 2, 3}) = 9, v({4, 5}) = 4, v({2, 4}) = 7
  – then (({1, 2, 3}, {4, 5}), (3, 3, 3, 3, 1)) is NOT in the core



Ice-Cream Game: Core

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

w = 500	w = 750	w = 1000
p = \$7	p = \$9	p = \$11

- v(Ø) = v({C}) = v({M}) = v({P}) = 0, v({C, M, P}) = 750
- v({C, M}) = 500, v({C, P}) = 500, v({M, P}) = 0
- (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!

Games with Empty Core

- The core is a very attractive solution concept
- However, some games have empty cores
- $G = (N, v)$
  - $N = \{1, 2, 3\}$, $v(C) = 1$ if $|C| > 1$ and $v(C) = 0$ otherwise
  - consider an outcome $(CS, \underline{\mathbf{x}})$
  - if $CS = (\{1\}, \{2\}, \{3\})$, the grand coalition can deviate
  - if $CS = (\{1, 2\}, \{3\})$, either 1 or 2 gets less than 1, so can deviate with 3
  - same argument for $CS = (\{1, 3\}, \{2\})$ or $CS = (\{2, 3\}, \{1\})$
  - suppose $CS = \{1, 2, 3\}$:

    $x_i > 0$ for some i, so $x(N\backslash\{i\}) < 1$, yet $v(N\backslash\{i\}) = 1$

Coaltion Fairness

- We will now define 2 more solution concepts:
  - the Shapley value
  - the Banzhaf index

  → fairness considerations

- For simplicity, we will define all these solution concepts for superadditive games only
  - however, all definitions generalize to non-superadditive games

Stability vs. Fairness

- Outcomes in the core may be unfair
- $G = (N, v)$
  - $N = \{1, 2\}$, $v(\emptyset) = 0$, $v(\{1\}) = v(\{2\}) = 5$, $v(\{1, 2\}) = 20$
- $(15, 5)$ is in the core:
  - player 2 cannot benefit by deviating
- However, this is unfair since 1 and 2 are symmetric
- How do we divide payoffs in a fair way?

Marginal Contribution

- A fair payment scheme would reward each agent according to his contribution
- First attempt: given a game $G = (N, v)$, set $x_i = v(\{1, ..., i-1, i\}) - v(\{1, ..., i-1\})$
  - payoff to each player = his marginal contribution to the coalition of his predecessors
- We have $x_1 + ... + x_n = v(N)$
  - **x** is a payoff vector
- However, payoff to each player depends on the order
- $G = (N, v)$
  - $N = \{1, 2\}$, $v(\emptyset) = 0$, $v(\{1\}) = v(\{2\}) = 5$, $v(\{1, 2\}) = 20$
  - $x_1 = v(1) - v(\emptyset) = 5$, $x_2 = v(\{1, 2\}) - v(\{1\}) = 15$

Average Marginal Contribution

- Idea: to remove the dependence on ordering, can average over all possible orderings
- $G = (N, v)$
  - $N = \{1, 2\}$, $v(\emptyset) = 0$, $v(\{1\}) = v(\{2\}) = 5$, $v(\{1, 2\}) = 20$
  - 1, 2: $x_1 = v(1) - v(\emptyset) = 5$, $x_2 = v(\{1, 2\}) - v(\{1\}) = 15$
  - 2, 1: $y_2 = v(2) - v(\emptyset) = 5$, $y_1 = v(\{1, 2\}) - v(\{2\}) = 15$
  - $z_1 = (x_1 + y_1)/2 = 10$, $z_2 = (x_2 + y_2)/2 = 10$
  - the resulting outcome is fair!
- Can we generalize this idea?

Shapley Value

- Reminder: a permutation of $\{1,\ldots, n\}$ is a one-to-one mapping from $\{1,\ldots, n\}$ to itself
  - let $P(N)$ denote the set of all permutations of N
- Let $S_\pi(i)$ denote the set of predecessors of i in $\pi \in P(N)$

$S_\pi(i)$	i	...

- For $C \subseteq N$, let $\delta_i(C) = v(C \cup \{i\}) - v(C)$
- Definition: the Shapley value of player i in a game $G = (N, v)$ with $|N| = n$ is

$$\phi_i(G) = 1/n! \sum_{\pi:\, \pi \in P(N)} \delta_i(S_\pi(i))$$

- In the previous slide we have $\phi_1 = \phi_2 = 10$

Shapley Value: Probabilistic Interpretation

- ϕ_i is i's average marginal contribution to the coalition of its predecessors, over all permutations
- Suppose that we choose a permutation of players uniformly at random, among all possible permutations of N
  - then ϕ_i is the expected marginal contribution of player i to the coalition of his predecessors

Shapley Value: Properties (1)-(2)

- Proposition: in any game G,

$$\phi_1 + \ldots + \phi_n = v(N)$$

  - $(\phi_1, \ldots, \phi_n)$ is a payoff vector
- Definition: a player i is a dummy in a game $G = (N, v)$ if $v(C) = v(C \cup \{i\})$ for any $C \subseteq N$
- Proposition: if a player i is a dummy in a game $G = (N, v)$ then $\phi_i = 0$

Shapley Value: Properties (3)-(4)

- <u>Definition</u>: given a game $G = (N, v)$,
  two players i and j are said to be symmetric
  if $v(C \cup \{i\}) = v(C \cup \{j\})$ for any $C \subseteq N \setminus \{i, j\}$
- <u>Proposition</u>: if i and j are symmetric then $\phi_i = \phi_j$
- <u>Definition</u>: Let $G_1 = (N, u)$ and $G_2 = (N, v)$ be two games with the same set of players.
  Then $G = G_1 + G_2$ is the game with the set of players N and characteristic function w given by
  $w(C) = u(C) + v(C)$ for all $C \subseteq N$
- <u>Proposition</u>: $\phi_i(G_1+G_2) = \phi_i(G_1) + \phi_i(G_2)$

Axiomatic Characterization

- Properties of Shapley value:
  1. Efficiency: $\phi_1 + \ldots + \phi_n = v(N)$
  2. Dummy: if i is a dummy, $\phi_i = 0$
  3. Symmetry: if i and j are symmetric, $\phi_i = \phi_j$
  4. Additivity: $\phi_i(G_1+G_2) = \phi_i(G_1) + \phi_i(G_2)$
- <u>Theorem</u>: Shapley value is the only payoff distribution scheme that has properties (1) - (4)

Banzhaf Index

- The difference between the Shapley value and the Banzhaf index can be described in terms of the underlying coalition formation model:
  - The **Shapley value** measures the agent's expected marginal contribution if agents join the coalition one by one in a random order
  - the **Banzhaf index** measures the agent's expected marginal contribution if each agent decides whether to join the coalition independently with probability 1/2

Banzhaf Index

- Instead of averaging over all permutations of players, we can average over all coalitions
- Definition: the Banzhaf index of player i in a game $G = (N, v)$ with $|N| = n$ is

$$\beta_i(G) = 1/2^{n-1} \sum_{C \subseteq N \setminus \{i\}} \delta_i(C) =$$
$$1/2^{n-1} \sum_{C \subseteq N \setminus \{i\}} v(C \cup \{i\}) - v(C)$$

- Satisfies dummy axiom, symmetry and additivity
- However, may fail efficiency: it may happen that $\sum_{i \in N} \beta_i \neq v(N)$

Shapley and Banzhaf: Examples

- Example 1 (unanimity game):
  - $G = (N, v)$, $|N| = n$, $v(C) = 1$ if $C = N$, $v(C) = 0$ otherwise
  - $\delta_i(C) = 1$ iff $C = N\setminus\{i\}$
  - $\phi_i(G) = (n-1)!/n! = 1/n$ for $i = 1, \ldots., n$
  - $\beta_i(G) = 1/2^{n-1}$ for $i = 1, \ldots., n$
- Example 2 (majority game):
  - $G = (N, v)$, $|N| = 2k$, $v(C) = 1$ if $|C| > k$, $v(C) = 0$ otherwise
  - $\delta_i(C) = 1$ iff $|C| = k$
  - $\phi_i(G) = (n-1)!/n! = 1/n$ for $i = 1, \ldots., n$
  - $\beta_i(G) = 1/2^{n-1} \times (2k)!/(k!)^2 \approx 2/\surd(\pi k)$ for $i = 1, \ldots., n$

4. Computational Issues in Coalitional Games

- We have defined many solution concepts - but can we compute them efficiently?
- Problem: the naive representation of a coalitional game is exponential in the number of players n
  – need to list values of all coalitions
- We are usually interested in algorithms whose running time is polynomial in n
- So what can we do?

How to Deal with Representation Issues?

- Strategy 1: oracle representation
  - assume that we have a black-box poly-time algorithm that, given a coalition $C \subseteq N$, outputs its value $v(C)$
  - for some special classes of games, this allows us compute some solution concepts using polynomially many queries
- Strategy 2: combinatorial games
  - consider games on combinatorial structures
  - problem: not all games can be represented in this way
- Strategy 3: give up on worst-case succinctness
  - devise complete representation languages that allow for compact representation of interesting games

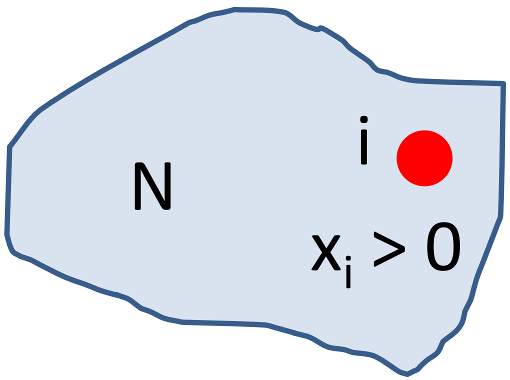
4.1 Oracle representations

Simple Games

- <u>Definition</u>: a game $G = (N, v)$ is simple if
  - $v(C) \in \{0, 1\}$ for any $C \subseteq N$
  - v is monotone: if $v(C) = 1$ and $C \subseteq D$, then $v(D) = 1$
- A coalition C in a simple game is said to be winning if $v(C) = 1$ and losing if $v(C) = 0$
- <u>Definition</u>: in a simple game, a player i is a veto player if $v(C) = 0$ for any $C \subseteq N \setminus \{i\}$
  - equivalently, by monotonicity, $v(N \setminus \{i\}) = 0$
- 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.

Simple Games: Characterization of the Core

- Proof ($\Leftarrow$):
  - suppose i is a veto player
  - consider a payoff vector $\underline{\mathbf{x}}$ with $x_i = 1$, $x_k = 0$ for $k \neq i$
  - no coalition C can deviate from $\underline{\mathbf{x}}$:
    - if $i \in C$, we have $\Sigma_{k\in C}\, x_k = 1 \geq v(C)$
    - if $i \notin C$, we have $v(C) = 0$



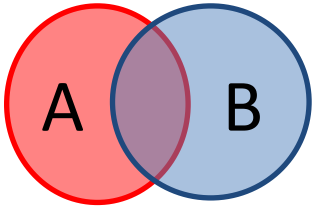
- Proof ($\Rightarrow$):
  - consider an arbitrary payoff vector $\underline{\mathbf{x}}$:
  - we have $\Sigma_{k\in N}\, x_k = v(N) = 1$; thus $x_i > 0$ for some $i \in N$
  - but then $N\setminus\{i\}$ can deviate:
    - since i is not a veto, $v(N\setminus\{i\}) = 1$, yet $x(N\setminus\{i\}) = 1 - x_i < 1$

Simple Games: Checking Non-Emptiness of the Core

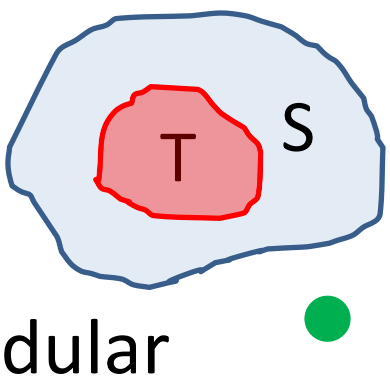
- Corollary: in a simple game G, a payoff vector **x** is in the core iff $x_i = 0$ for any non-veto player i
  - proved similarly
- Checking if a player i is a veto player is easy
  - a single oracle access to compute $v(N\setminus\{i\})$
- Thus, in simple games
  - checking non-emptiness of the core or
  - checking if a given outcome is in the core

  is easy given oracle access to the characteristic function
  - this is no longer the case if we allow coalition structures

Convex Games



- Definition: a function $f{:}2^N \to R$ is called supermodular
if $f(\emptyset) = 0$ and $f(A \cup B) + f(A \cap B) \geq f(A) + f(B)$
for any $A, B \subseteq N$ (not necessarily disjoint)
  - any supermodular function is superadditive, but the converse is not true
- Proposition: if f is supermodular , $T \subset S$, and $i \notin S$, then $f(T \cup \{i\}) - f(T) \leq f(S \cup \{i\}) - f(S)$
  - a player is more useful when he joins a bigger coalition



- Definition: a game $G = (N, v)$ is convex if its characteristic function is supermodular

Convex Games: Non-Emptiness of The Core

- Proposition: any convex game has a non-empty core
- Proof:
  - set $x_1 = v(\{1\})$,
    $x_2 = v(\{1, 2\}) - v(\{1\})$,
    ...
    $x_n = v(N) - v(N \setminus \{n\})$
    - i.e., pay each player his marginal contribution to the coalition formed by his predecessors
  - **x** is a payoff vector: $x_1 + x_2 + \ldots + x_n =$
    $= v(\{1\}) + v(\{1, 2\}) - v(\{1\}) + \ldots + v(N) - v(N \setminus \{n\}) = v(N)$
  - remains to show that $(x_1, x_2, \ldots, x_n)$ is in the core

Convex Games Have Non-Empty Core

- Proof (continued):
  - $x_1 = v(\{1\})$, $x_2 = v(\{1, 2\}) - v(\{1\})$, ..., $x_n = v(N)-v(N\setminus\{n\})$
  - pick any coalition $C = \{i, j, ..., s\}$, where $i < j < ... < s$
  - we will prove $v(C) \le x_i + x_j + ... + x_s$, i.e., C cannot deviate
  - $v(C) = v(\{i\}) + v(\{i, j\}) - v(\{i\}) + ... + v(C) - v(C\setminus\{s\})$
    - $v(\{i\}) = v(\{i\}) - v(\emptyset) \le v(\{1, ..., i-1, i\}) - v(\{1, ..., i-1\}) = x_i$
    - $v(\{i, j\}) - v(\{i\}) \le v(\{1, ..., j-1, j\}) - v(\{1, ..., j-1\}) = x_j$
    - ....
    - $v(C) - v(C\setminus\{s\}) \le v(\{1, ..., s-1, s\}) - v(\{1, ..., s-1\}) = x_s$
  - thus, $v(C) \le x_i + x_j + ... + x_s$

	i			j		s	

Convex Games: Remarks

- This proof suggests a simple algorithm for constructing an outcome in the core
  - order the players as 1, ..., n
  - query the oracle for v({1}), v({1, 2}), ..., v(N)
  - set $x_i = v(\{1, ..., i-1, i\}) - v(\{1, ..., i-1\})$
- This argument also shows that for convex games the Shapley value is in the core
  - the core is a convex set
  - Shapley value is a convex combination of outcomes in the core

Checking Non-emptiness of the Core: Superadditive Games

- An outcome in the core of a superadditive game satisfies the following constraints:

  $x_i \geq 0$ for all $i \in N$

  $\Sigma_{i \in N}\, x_i = v(N)$

  $\Sigma_{i \in C}\, x_i \geq v(C)$ for any $C \subseteq N$

- A linear feasibility program, with one constraint for each coalition: 2^n+n+1 constraints
  - sometimes can be solved in polynomial time solvers using separation oracles

Superadditive Games: Computing the Least Core

- LFP for the core ⇨ LP for the least core

  min ε

  $x_i \geq 0$ for all $i \in N$

  $\Sigma_{i \in N}\, x_i = v(N)$

  $\Sigma_{i \in C}\, x_i \geq v(C) - \varepsilon$ for any $C \subseteq N$

- A minimization program, rather than a feasibility program
  - sometimes can be solved in polynomial time using a separation oracle

Core and Related Concepts: Non-Superadditive Games

- What if the game is not superadditive?
- Can solve a similar LFP for each coalition structure $CS = (C^1, \ldots, C^k)$:

  $x_i \geq 0$ for all $i \in N$

  $\Sigma_{i \in C^1} x_i = v(C^1)$

  ...

  $\Sigma_{i \in C^k} x_i = v(C^k)$

  $\Sigma_{i \in C} x_i \geq v(C)$ for any $C \subseteq N$
- Running time: # of partitions of N x time to solve an exp-sized LFP - infeasible in general.

4.2 Combinatorial optimization games

Weighted Voting Games

- n parties in the parliament
- Party i has w_i representatives
- A coalition of parties can form a government only if its total size is at least q
  - usually $q \geq \lfloor \Sigma_{i=1, \dots, n} w_i /2 \rfloor + 1$: strict majority
- Notation: $w(C) = \Sigma_{i \in C} w_i$
- This setting can be described by a game $G = (N, v)$, where
  - $N = \{1, \dots, n\}$
  - $v(C) = 1$ if $w(C) \geq q$ and $v(C) = 0$ otherwise
- Observe that weighted voting games are simple games
- Notation: $G = [q; w_1, \dots, w_n]$
  - q is called the quota

Weighted Voting Games: UK

- United Kingdom, 2005:
  - 650 seats, $q = 326$
  - Conservatives (C): 196
  - Labour (L): 354
  - Liberal Democrats (LD): 62
  - 8 other parties (O), with a total of 38 seats

- $N = \{C, L, LD, O\}$
- for any $X \subseteq N$, $v(X) = 1$ if and only if $L \in X$
- L is a veto player, C, LD, and O are dummies
- $\phi_L = 1$, $\phi_C = \phi_{LD} = \phi_O = 0$

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

Weighted Voting Games as Resource Allocation Games

- Each agent i has a certain amount of a resource w_i
  - time or money or battery power
- One or more tasks with a resource requirement q and a value V
- If a coalition has enough resources to complete the task (q or more units), it earns its value V, else it earns 0
  - By normalization, can assume $V = 1$
- If $q < \Sigma_i w_i/2$, grand coalition need not form
  - weighted voting games with coalition structures

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

Weighted Voting Games: Computational Aspects

- Deciding if a player is a dummy: coNP-complete
- Computing Shapley value and Banzhaf index:
  - #P-complete [Deng & Papadimitriou'94]
  - hard to approximate
- Computing the core/checking if an outcome is in the core:
  - poly-time (since WVG are simple games)
  - if we allow coalition structures, these problems become computationally hard [Elkind et al.'08b]

Weighted Voting Games: Small Weights

- Suppose all weights are at most polynomial in n
  - realistic in many applications
- Then
  - Shapley value and Banzhaf index can be computed in poly-time by dynamic programming [Matsui & Matsui'00]
  - value of the least core is poly-time computable [Elkind et al.'09a]
  - nucleolus is poly-time computable [Elkind and Pasechnik'09]

WVG and Simple Games

- WVGs are simple games
- Can every simple game be represented as a WVG?
- $G = (N, v)$:
  - $N = \{1, 2, 3, 4\}$
  - $v(C) = 1$ iff $C \cap \{1, 3\} \neq \emptyset$ and $C \cap \{2, 4\} \neq \emptyset$
- Suppose $G = [q; w_1, w_2, w_3, w_4]$

$w_1 + w_2 \geq q, \quad w_3 + w_4 \geq q$

$$\left.\begin{array}{l} w_1 + w_2 + w_3 + w_4 \geq 2q \\ w_1 + w_3 < q, \quad w_2 + w_4 < q \\ w_1 + w_2 + w_3 + w_4 < 2q \end{array}\right\} \text{a contradiction!}$$

A Generalization: Vector Weighted Voting Games

- The game in the previous slide can be thought of as a combination of two WVGs:
  - $G^{odd} = [1; 1, 0, 1, 0]$ and $G^{even} = [1; 0, 1, 0, 1]$
  - to win, a coalition needs to win in both games
- <u>Definition</u>: a k-weighted voting game is a tuple $[N; \underline{\mathbf{q}}; \underline{\mathbf{w}}_1, \dots, \underline{\mathbf{w}}_n]$, where $|N| = n$ and
  - $\underline{\mathbf{q}} = (q^1, \dots, q^k)$ is a vector of k real quotas
  - for each $i \in N$, $\underline{\mathbf{w}}_i = (w^1_i, \dots, w^k_i)$ is a vector of k real weights
- $v(C) = 1$ if $\sum_{i \in C} w_i^{\,j} \geq q^{\,j}$ for each $j = 1, \dots, k$ and $v(C) = 0$ otherwise

Vector Weighted Voting Games

- Given a k-VWVG $G = [N; \underline{\mathbf{q}}; \underline{\mathbf{w}}_1, \ldots, \underline{\mathbf{w}}_n]$, we can define $G^j = [q^j; w^j_1, \ldots, w^j_n]$
- G^j is a weighted voting game
  - we will refer to G^j as the j-th component of G
- To win in G, a coalition needs to win in each of the component games
  - we can write $G = G^1 \wedge \ldots \wedge G^k$
  - thus, G is a conjunction of its component games
- a k-VWG models a resource allocation games with k types of resources
  - each task needs q^j units of resource j

VWVG in the Wild: EU Voting

- Voting in the European Union is a 3-WVG
  $G = G^1 \wedge G^2 \wedge G^3$, where
  - G^1 corresponds to commissioners
  - G^2 corresponds to countries
  - G^3 corresponds to population

- The players are the 27 member states: Germany, UK, France, Italy, Spain, Poland, Romania, The Netherlands, Greece, Czech Republic, Belgium, Hungary, Portugal, Sweden, Bulgaria, Austria, Slovak Republic, Denmark, Finland, Ireland, Lithuania, Latvia, Slovenia, Estonia, Cyprus, Luxembourg, Malta.

EU Voting Game

- G^1 = [255; 29, 29, 29, 29, 27, 27, 14, 13, 12, 12, 12, 12, 12, 10, 10, 10, 7, 7, 7, 7, 7, 4, 4, 4, 4, 4, 3]
- G^2 = [14; 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
- G^3 = [620; 170, 123, 122, 120, 82, 80, 47, 33, 22, 21, 21, 21, 21, 18, 17, 17, 11, 11, 11, 8, 8, 5, 4, 3, 2, 1, 1]
  - UK, Greece, Estonia
- For a proposal to pass, it needs to be supported by
  - 74% of the commissioners
  - 50% of the member states
  - 62% of the EU population

VWVGs and Simple Games

- VWVGs are strictly more expressive than WVGs
- <u>Theorem</u>: any simple game can be represented as a vector weighted voting game
- Proof: consider a simple game $G=(N, v)$
  - for each losing coalition $C \subseteq N$, we construct a game $G^C = [q^C; w^C_1, \ldots, w^C_n]$ as follows:

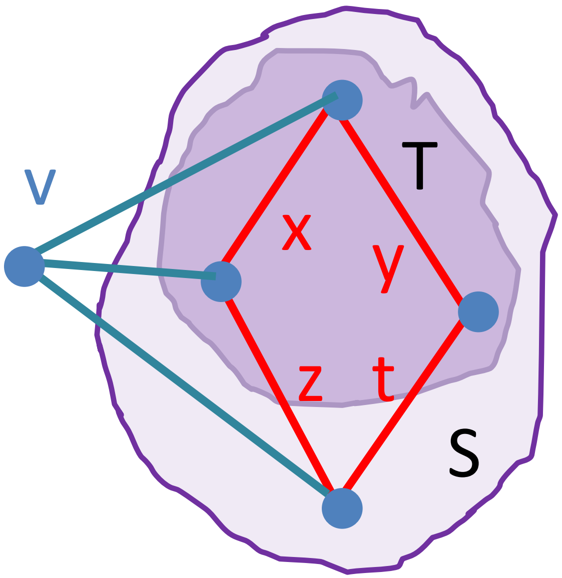
    $q^C = 1$, $w^C_i = 1$ if $i \notin C$ and $w^C_i = 0$ if $i \in C$
    - D loses in G^C iff $D \subseteq C$
  - Let $G^* = \Lambda_{v(C) = 0}\, G^C$
  - if $v(D) = 0$, D loses in G^D and hence in G^*
  - if $v(D) = 1$, by monotonicity D wins in each component game and hence in G^*

Dimensionality

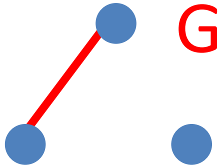
- Vector weighted voting games form a complete representation language for simple games
- However, the construction in the previous slide may use exponentially many component games
- Definition: the dimension dim(G) of a simple game G is the minimum number of components in its VWVG representation
  - every simple game has dimension $O(2^n)$
  - there exist simple games of dimension $\Omega(2^{n/2-1})$

Induced Subgraph Games

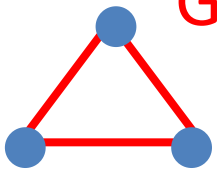
- Players are vertices of a weighted graph
- Value of a coalition = total weight of internal edges
  - $v(T) = x+y$, $v(S) = x+y+z+t$
- Models social networks
  - Facebook, LinkedIn
  - cell phone companies with free in-network calls
- If all edge weights are non-negative, this game is convex:
  - $\delta_v(S) \geq \delta_v(T)$

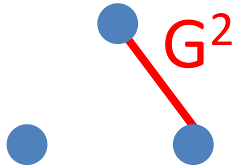


Induced Subgraph Games: Complexity
[Deng, Papadimitriou'94]

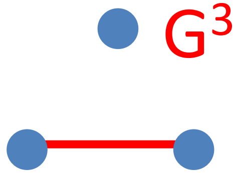

G^1

- If all edge weights are non-negative, the core is non-empty
  - also, we can check in poly-time if a given outcome is in the core
- In general, determining emptiness of the core is NP-complete


G

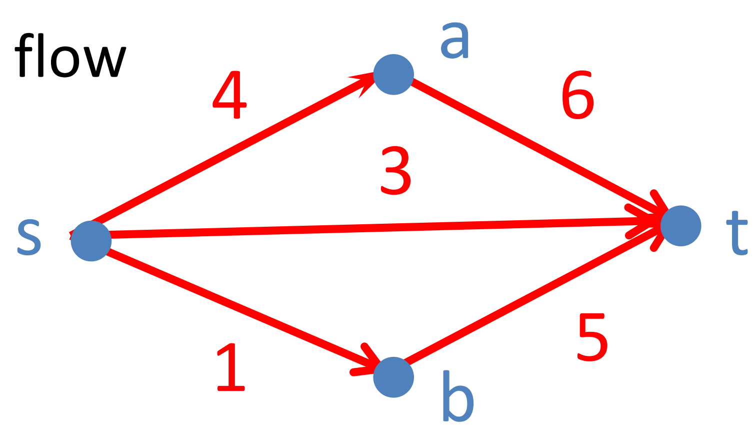

G^2

- Shapley value is easy to compute:
  - let $E = \{e^1, \dots, e^k\}$ be the list of edges of the graph
  - let G^j be the induced subgraph game on the graph that contains edge e^j only
  - we have $G = G^1 + \dots + G^k$
  - $\phi_i(G^j) = w(e^j)/2$ if e^j is adjacent to i and 0 otherwise
  - $\phi_i(G)$ = (weight of edges adjacent to i)/2

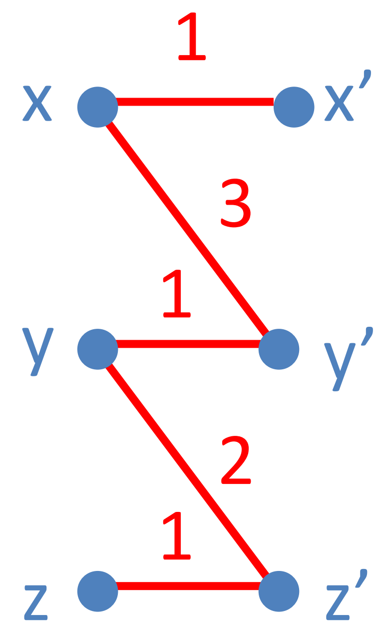

G^3

Network Flow Games

- Agents are edges in a network with source s and sink t
  - edge e_i has capacity c_i
- Value of a coalition = amount of s–t flow it can carry
  - v({sa, at}) = 4, v({sa, at, st}) = 7
- Thresholded network flow games (TNFG): there exists a threshold T such that
  - v(C) = 1 if C can carry ≥ T units of flow
  - v(C) = 0 otherwise
- TNFG with T = 6
  - v({sa, at}) = 0, v({sa, at, st}) = 1



Assignment Games
[Shapley & Shubik'72]



- Players are vertices of a bipartite graph (V, W, E)
- Value of a coalition = weight of the max-weight induced matching
  - v({x, y, z}) = 0, v({x, x', y'}) = 3
- Generalization: matching games
  - same definition, but the graph need not be bipartite

4.3 Complete representation languages

Coalitional Skill Games
[Bachrach & Rosenschein'08]

- Set of skills $S = \{s_1, . . . , s_k\}$
- Set of agents N: agent i has a subset of skills $S_i \subseteq S$
- Set of tasks $T = \{t_1, . . . , t_m\}$
  - each task t_j requires a subset of skills $S(t_j) \subseteq S$
- A skill set of a coalition C: $s(C) = \cup_{i \in C} S_i$
- Tasks that C can perform: $T(C) = \{t_j \mid S(t_j) \subseteq S(C)\}$
- Utility function $u : 2^T \to \mathbb{R}$
  - e.g., sum or max of values of individual tasks
- Characteristic function: $v(C) = u(T(C))$

Coalitional Skill Games: Expressiveness and Complexity

- Any monotone game can be expressed as a CSG:
  - given a game $G = (N, v)$,
    we create a task t^C and set $u(t^C) = v(C)$ for any $C \subseteq N$
  - each agent i has a unique skill s_i
  - t^C requires the skills of all agents in C
  - set $u(T') = \max \{ u(t) \mid t \in T' \}$
  - $u(T(C)) = \max \{u(t^D) \mid D \subseteq C\} = \max \{v(D) \mid D \subseteq C\} = v(C)$
- However, the representation is only succinct when the game is naturally defined via a small set of tasks
- [Bachrach&Rosenschein'08] discuss complexity of many solution concepts under this formalism

Synergy Coalition Games [Conitzer & Sandholm'06]

- Superadditive game: $v(C \cup D) \geq v(C) + v(D)$ for any two disjoint coalitions C and D
- Idea: if a game is superadditive, and $v(C) = v(C_1) + \ldots + v(C_k)$ for any partition $(C_1, \ldots, C_k)$ of C (no synergy), no need to store $v(C)$
- Representation: list $v(\{1\})$, ... $v(\{n\})$ and all synergies
- Succinct when there are few synergies
- This representation allows for efficient checking if an outcome is in the core.
- However, it is still hard to check if the core is non-empty.

Marginal Contribution Nets
[Ieong & Shoham'05]

- Idea: represent the game by a set of rules of the form pattern $\rightarrow$ value
  - pattern is a Boolean formula over N
  - value is a number
- A rule applies to a coalition if its fits the pattern
- v(C) = sum of values of all rules that apply to C
- Example:

R_1: $(1 \wedge 2) \vee 5 \rightarrow 3$

R_2: $2 \wedge 3 \rightarrow -2$

$v(\{1, 2\}) = 3$, $v(\{2, 3\}) = -2$, $v(\{1, 2, 3\}) = 1$

Marginal Contribution Nets

- Computing the Shapley value:
  - let $G(R_1, \ldots, R_k)$ be the game given by the set of rules $R_1, \ldots, R_k$
  - we have $G(R_1, \ldots, R_k) = G(R_1) + \ldots + G(R_k)$
  - thus, by additivity it suffices to compute players' Shapley values in games with a single rule R
  - if $R = \psi \rightarrow x$, where ψ is a conjunction of k variables, then $\phi_i = x/k$ if i appears in ψ and 0 otherwise
  - a more complicated (but still poly-time) algorithm for read-once formulas [Elkind et al.'09b]
  - NP-hard for if ψ is an arbitrary Boolean formula
- Core-related questions are computationally hard [Ieong&Shoham'05]

Slides based on the ones elaborated by

- Edith Elkind
  (Nanyang Technological University, Singapore)
- Talal Rahwan

  (University of Southampton, UK)
- Nicholas R. Jennings

  (University of Southampton, UK)