Game Theory for Computer Science

Michael Wooldridge

Department of Computer Science
University of Oxford
United Kingdom

mjw@cs.ox.ac.uk

Overview

1. What is game theory?
2. Why is game theory relevant to computer science/AI?
3. Preferences and Utility
4. Non-cooperative games
5. Cooperative games
6. Further reading
7. History of game theory

Part I

What is game theory?

What is Game Theory?

- The mathematical theory of interaction between self-interested agents ("players").
- Self-interest: Players assumed to act in their own interests, in pursuit of their preferences
- Focus on decision-making where each player's decision can influence the outcomes (and hence well-being) of other players.
- Each player must consider how each other player will act in order to make its optimal choice: hence strategic considerations
- If all players have the same preferences, then game theoretic analysis is essentially redundant: there is common purpose.
- If a system has one designer, or is "owned" by a single individual, we can usually assume common purpose.

What is a "Game"?

- A "game" in the sense of game theory is an abstract model of a particular scenario in which self-interested players interact.
- Abstract in the sense that we only include detail relevant to the decisions that players make:
  - leads to claims that game theoretic models are "toy"
  - aim is to isolate issues that are central to decision making.
- Game theory origins: study of parlor games (e.g., chess)
  - such games are useful for highlighting key concepts
  - but the term "game" conveys something trivial :-(

Solution Concepts

- Key concern in game theory is to understand what the outcomes of a game will be, under the assumption that players act rationally (in their best interests).
- But it is often not clear what the best thing to do is.
- Solution concepts attempt to characterise rational outcomes of games
- For every game, a solution concept identifies a subset of the outcomes of the game – those that would occur if players acted according to the corresponding model of rational choice
- Problems . . . what happens if the solution concept says:
  - there is no rational outcome?
  - there are multiple rational outcomes?

- Under a descriptive interpretation, we take game theory as predicting how people will act in strategic settings, and explaining why they acted the way they did.
- A major area of research to determine the extent to which game theoretic solution concepts predict human choices (somewhat controversial)

Interpreting Game Theory

Binmore[1] on when descriptive interpretations work

- In real life settings, social norms (and in particular, norms of cooperation) often play a part in how people make decisions. However, if the incentives are sufficiently large, then these can override such norms.
- For incentives (such as payments) to influence behavior, they must be adequate.
- For players to make rational choices, the game they are playing must be sufficiently simple.
- Players will adapt their behavior over time towards more rational outcomes, if they are given sufficient opportunity for trial-and-error learning.

[1] Ken Binmore, *Does Game Theory Work?*, MIT Press, 2007.

Interpreting Game Theory

Normative Interpretations

- Under a normative interpretation, we take game theory as giving us advice: telling us what we ought to do in a real-world situation.
- Whether the advice is useful depends on whether the game model used was appropriate, and whether the assumptions on which the model depends are satisfied. (Typical assumptions: everybody knows everybody's preferences, actions, and their consequences, everybody acts rationally,...)
- Game theory can be used to design interaction scenarios: (mechanism design).
  EXAMPLE 1. 3G spectrum auctions in 2000 yielded $35 billion for UK government.
  EXAMPLE 2. "security games" paradigm (Milind Tambe)

Non-Cooperative *versus* Cooperative Games

- Game theory is usually sub-divided into non-cooperative and cooperative versions.
- Non-cooperative game theory is bigger and better-known: it concerns settings where players must act alone. Solution concepts in non-cooperative game theory relate to individual action.
- Cooperative game theory is concerned with settings where players can make binding agreements to work together, allowing for teamwork, cooperation, joint action.

Part II

Why is game theory relevant to computer science?

Mechanisms and Protocols

- Distributed systems research has focussed on protocols (TCP/IP, leader election, bluetooth, . . .)
  Typical issues: deadlock, mutual exclusion. . .
- In multi-agent systems, we study mechanisms.

  mechanism = protocol + self interest

- Mechanisms take into account the fact that protocol participants are not benevolent entities – they are self-interested.

  strategic considerations come to the fore.

- Treating mechanisms as if they were simply protocols misses a big part of the story.
  example: sniping on eBay
- In multi-agent systems, mechanism participants are software agents.

Two perspectives on game theory in computer science

- Algorithmic mechanism design: take economic factors (in particular: self interested behaviour) into consideration when designing computational systems.
- Electronic market design: Use computer science techniques in the design of economic systems.

Computational issues for game theory

- Let Γ be a class of games. (It doesn't matter exactly what the games $G \in \Gamma$ are.)
- Associated with Γ is a set Ω of outcomes.
- Where $G \in \Gamma$ is a specific game, let Ω_G denote the possible outcomes of G.
- A solution concept σ for a class of games Γ with outcomes Ω as a function:

  $$\sigma : \Gamma \to 2^{\Omega}$$

  such that $\sigma(G) \subseteq \Omega_G$.

Computational issues for game theory

Non-emptiness: Given $G \in \Gamma$, is it the case that $\sigma(G) \neq \emptyset$?
Thus, non-emptiness simply asks whether the game has any outcome that is rational according to the solution concept σ.

Membership: Given $G \in \Gamma$ and $\omega \in \Omega_G$, is it the case that $\omega \in \sigma(G)$?
Asks whether a given outcome is rational according to σ.

Computation: Given $G \in \Gamma$, output some ω such that $\omega \in \sigma(G)$.
Here, we actually want to compute a rational outcome of the game.

Part III

Non-cooperative games

Players and Outcomes

- Throughout this lecture, the set of players is denoted by $N = \{1, \ldots, n\}$
- Assume $\Omega = \{\omega_1, \omega_2, \ldots\}$ is the set of "outcomes". These are the consequences of player's choices.
- Ω may be:
  - all the possible outcomes of a game of chess
  - the possible outcomes of negotiations between nations
  - the possible outcomes of an eBay auction
  - … and so on.

Preference Relations

A preference relation for player $i \in N$ is a binary relation $\succeq_i \; \subseteq \Omega \times \Omega$, which is required to satisfy:

1. Reflexivity:
   $\omega \succeq_i \omega$ for all $\omega \in \Omega$
2. Totality:
   for all $\omega, \omega' \in \Omega$ we have either $\omega \succeq_i \omega'$ or $\omega' \succeq_i \omega$.
3. Transitivity:
   for all $\omega, \omega', \omega'' \in \Omega$, if $\omega \succeq_i \omega'$ and $\omega' \succeq_i \omega''$ then $\omega \succeq_i \omega''$.

Indifference and Strict Preference

Indifference:

If both $\omega \succeq_i \omega'$ and $\omega' \succeq_i \omega$ then we say i is indifferent between ω and ω', and write

$$\omega \sim_i \omega'$$

Strict Preference:

If $\omega \succeq_i \omega'$ but not $\omega' \succeq_i \omega$ then we say i strictly prefers ω over ω' and write

$$\omega \succ_i \omega'$$

Interpeting Preferences (IMPORTANT)

- $\omega \succ_i \omega'$ means that:
  - if you have a choice between ω and ω', you will always choose ω
  - if you have two actions available, one of which will bring about ω, the other of which will bring about ω', you will always perform ω
- Notice that preference is interpretated wrt your behaviour
- Player *i*'s preference relation must capture everything about the game that player *i* cares about.
  For example if player *i* cares about other players, then this is reflected in his preferences. (Many arguments in game theory would be avoided if everybody understood this!)

Utility functions

- It is useful to represent preference relations by attaching numbers to outcomes: higher numbers are more preferred.
- The numbers are called utility values or utilities.
- A utility function $u_i : \Omega \to \mathbb{R}$ is said to represent player i's preference relation $\succeq_i$ iff we have:

$$\begin{aligned} u_i(\omega) \geq u_i(\omega') &\quad \text{iff} \quad \omega \succeq_i \omega' \\ u_i(\omega) > u_i(\omega') &\quad \text{iff} \quad \omega \succ_i \omega' \end{aligned}$$

Theorem

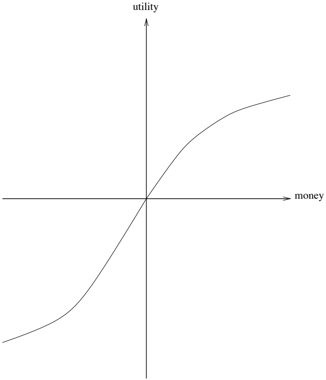
For every preference relation $\succeq_i \; \subseteq \Omega \times \Omega$ there is a utility function $u_i : \Omega \to \mathbb{R}$ that represents $\succeq_i$.

What is Utility?

- We use (numeric) utility values because it allows us to use numeric techniques for solving games.
- Utilities are selected simply to represent preference relations $\succeq_i$.
- It is a fallacy to claim you choose ω over ω' because $u_i(\omega) > u_i(\omega')$ – you make this choice because $\omega \succ_i \omega'$. The u_i values were chosen to reflect this.
- But, if we picked the numbers right, then you behave as though you were maximising utility.
- Utility values don't represent intensity: they are ordinal values, which indicate relative rankings.
- Interpersonal comparisons of utility are difficult. "One util" for me is not the same as "one util" for you.

Utility is not money!

- Much misunderstanding caused by people interpreting utility as money, leading to the implication that game theory is about “greed”. . .
- Utility as money is often a useful analogy.
- For the record, a typical relationship between utility & money:



Game Forms

- We now introduce our first game model.
- Players simultaneously choose a strategy, and as a result of the combination of strategies selected, an outcome in Ω will result;
- Player i's strategies given in a set Σ_i, with members σ_i etc.
- Environment behaviour given by outcome function:

$$g : \Sigma_1 \times \cdots \times \Sigma_n \to \Omega$$

- A game form is a structure:

$$\langle N, \Omega, \Sigma_1, \ldots, \Sigma_n, g \rangle$$

An Example Game Form

- Suppose we have $N = \{1, 2\}$, $\Omega = \{\omega_1, \ldots, \omega_4\}$, and $\Sigma_1 = \Sigma_2 = \{C, D\}$. (Read the strategies as C = cooperate, D = defect.)
- Here is an outcome function:

$$g(D, D) = \omega_1 \quad g(D, C) = \omega_2 \quad g(C, D) = \omega_3 \quad g(C, C) = \omega_4$$

- This game form is sensitive to actions of both agents.

Another Game Form

$$g(D, D) = \omega_1 \quad g(D, C) = \omega_1 \quad g(C, D) = \omega_1 \quad g(C, C) = \omega_1$$

Neither agent has any influence in this environment.

Yet Another Game Form

$$g(D, D) = \omega_1 \quad g(D, C) = \omega_2 \quad g(C, D) = \omega_1 \quad g(C, C) = \omega_2$$

This environment is controlled by player 2.

Adding preferences

- Suppose we have the first case, where both agents can influence the outcome.
- Now suppose players have utility functions as follows:

$$\begin{array}{llll} u_1(\omega_1) = 1 & u_1(\omega_2) = 1 & u_1(\omega_3) = 4 & u_1(\omega_4) = 4 \\ u_2(\omega_1) = 1 & u_2(\omega_2) = 4 & u_2(\omega_3) = 1 & u_2(\omega_4) = 4 \end{array}$$

- With a bit of abuse of notation:

$$\begin{array}{llll} u_1(D,D) = 1 & u_1(D,C) = 1 & u_1(C,D) = 4 & u_1(C,C) = 4 \\ u_2(D,D) = 1 & u_2(D,C) = 4 & u_2(C,D) = 1 & u_2(C,C) = 4 \end{array}$$

- Sgent 1's preferences are:

$$(C,C) \sim_1 (C,D) \quad \succ_1 \quad D,C \sim_1 D,D$$

- Informally, "C" is the rational choice for 1. (Why?)
- In what follows, we drop the outcome function g and assume utility functions are of the form:

$$u_i : \Sigma_1 \times \cdots \times \Sigma_n \to \mathbb{R}$$

Normal Form Games

A normal form game is a structure:

$$\langle N, \Sigma_1, \ldots, \Sigma_n, u_1, \ldots, u_n \rangle$$

where:

- $N = \{1, \ldots, n\}$ is the set players;
- Σ_i is a set of possible strategies for player $i \in N$;
- $u_i : \Sigma_1 \times \cdots \times \Sigma_n \to \mathbb{R}$ is the utility function for agent $i \in N$.

Notice that the utility i gets depends not on only her actions, but on the actions of others, and similarly for other agents.
For i to find the best action involves deliberating about what others will do, taking into account the fact that they will also try to maximise their utility taking into account how i will act.

Payoff Matrices

- We can summarise the previous normal form game in a payoff matrix

		1	
		defect	coop
2	defect	1 (agent 1), 1 (agent 2)	4 (agent 1), 1 (agent 2)
	coop	1 (agent 1), 4 (agent 2)	4 (agent 1), 4 (agent 2)

- Agent 1 is the column player.
- Agent 2 is the row player.

Solution Concepts

- If players act rationally, what will the outcome of the game be?
- Answered in solution concepts:
  - dominant strategy;
  - Nash equilibrium strategy;
  - Pareto optimal strategies;
  - strategies that maximise social welfare.
- A key concept to understand these is the notion of best response.

Strategy Profiles

- A strategy profile, $\vec{\sigma}$, is a tuple of strategies, one for each player:
$$\vec{\sigma} = (\sigma_1, \ldots, \sigma_n).$$
- Where $\vec{\sigma}$ is a strategy profile and $\sigma_i \in \Sigma_i$, we denote the strategy profile obtained by replacing the i component with σ_i by $(\vec{\sigma}_{-i}, \sigma_i)$.

Dominant Strategies

- Strategy σ_i is dominant for player i if no matter what strategies other players chooses, i will do at least as well playing σ_i as it would doing anything else.
- To say that σ_i is a dominant strategy for i is to say that σ_i is a best response to all of its counterpart strategies
- A dominant strategy equilibrium is a strategy profile in which every player chooses a dominant strategy.
- A strong solution concept. . . but unfortunately, there isn't always a dominant strategy.

(Pure Strategy) Nash Equilibrium

- A strategy profile $\vec{\sigma}$ is a Nash equilibrium if no player would rather have done something else, assuming the other players stuck with their strategies.
- Formally, $\vec{\sigma}$ is a NE if there is no player $i \in N$ and strategy $\sigma_i' \in \Sigma_i$ such that

$$u_i(\vec{\sigma}_{-i}, \sigma_i') > u_i(\vec{\sigma}).$$

- Nobody can benefit by deviating from a Nash equilibrium.
- Unfortunately:
  1. Not every game has a (pure) NE.
  2. Some games have more than one NE (equilibrium selection problem)
  3. Some NE are bad!

The Matching Pennies Game

- Players 1 and 2 simultaneously choose the face of a coin, either “heads” or “tails”.
- If they show the same face, then 1 wins, while if they show different faces, then 2 wins.

Matching Pennies: The Payoff Matrix

	1 heads	1 tails
2 heads	1 -1	-1 1
2 tails	-1 1	1 -1

Mixed Strategies for Matching Pennies

- No pair of strategies forms a pure NE in matching pennies: whatever pair of strategies is chosen, somebody wishes they had done something else.
- The solution is to allow mixed strategies:
  - play “heads” with probability 0.5
  - play “tails” with probability 0.5.
- If both players do this, we have a NE strategy profile.

Mixed Strategies

A mixed strategy has the form

$$\begin{array}{l} \text{play } \sigma_1 \text{ with probability } p_1 \\ \text{play } \sigma_2 \text{ with probability } p_2 \\ \cdots \\ \text{play } \sigma_k \text{ with probability } p_k. \end{array}$$

which must satisfy the probability constraint:

$$p_1 + p_2 + \cdots + p_k = 1.$$

Nash's Theorem

Theorem

Every finite game has a Nash equilibrium in mixed strategies.

- Guarantees the existence of NE.
- But there may be more than one Nash equilibrium... the equilibrium selection problem.

Pareto Optimality

- An outcome is said to be Pareto optimal (or Pareto efficient) if there is no other outcome that makes one agent better off without making another agent worse off.
- If an outcome is Pareto optimal, then at least one agent will be reluctant to move away from it (because this agent will be worse off).
- If an outcome ω is not Pareto optimal, then there is another outcome ω' that makes everyone as happy, if not happier, than ω.
  "Reasonable" agents would agree to move to ω' in this case. (Even if I don't directly benefit from ω', you can benefit without me suffering.)

Social Welfare

- The social welfare of an outcome ω is the sum of the utilities that each agent gets from ω:

$$\sum_{i \in N} u_i(\omega)$$

- Think of it as the "total amount of money in the system".
- As a solution concept, may be appropriate when the whole system (all agents) has a single owner (then overall benefit of the system is important, not individuals).

Competitive and Zero-Sum Interactions

- Where preferences of agents are diametrically opposed we have strictly competitive scenarios.
- Zero-sum encounters are those where utilities sum to zero:

$$\sum_{i \in N} u_i(\omega) = 0 \qquad \text{for all } \omega \in \Omega.$$

- Zero sum encounters are bad news: for me to get +ve utility you have to get negative utility! The best outcome for me is the worst for you!
- Zero sum encounters in real life are very rare . . . but people frequently act as if they were in a zero sum game.

The Prisoner's Dilemma Game

"Two men are collectively charged with a crime and held in separate cells, with no way of meeting or communicating. They are told that:

- if one confesses and the other does not, the confessor will be freed, and the other will be jailed for three years;
- if both confess, then each will be jailed for two years.

Both prisoners know that if neither confesses, then they will each be jailed for one year."

Payoff matrix for the Prisoner's Dilemma

		1	
		defect	coop
2	defect	-2 (1), -2 (2)	-3 (1), 0 (2)
	coop	0 (1), -3 (2)	-1 (1), -1 (2)

- Top left: If both defect, then both get punishment for mutual defection: two years in jail.
- Top right: If 1 cooperates and 2 defects, 1 gets sucker's payoff (3 yrs jail) while 2 goes free.
- Bottom left: If 2 cooperates and 1 defects, 2 gets sucker's payoff, 1 goes free.
- Bottom right: Reward for mutual cooperation, 1 year in jail.

What Should You Do?

- Mutual defection (2 years in jail each) is a dominant strategy equilibrium:
  - Suppose he defects: my best response is to defect.
  - Suppose he cooperates: my best response is to defect.
- But intuition says this is not the best outcome:

  Surely they should both cooperate – then they only serve 1 year in jail!

Solution Concepts

- (D, D) is a dominant strategy equilibrium.
- (D, D) is the only Nash equilibrium.
- All outcomes except (D, D) are Pareto optimal.
- (C, C) maximises social welfare.

The Dilemma!

- This apparent paradox has been described as "a fundamental problem of multi-agent interactions".
- Real world examples:
  - nuclear arms reduction ("why don't I keep mine. . . ")
  - free rider systems — public transport;
  - in the UK — television licenses.
- The prisoner's dilemma is ubiquitous.
- Can we recover cooperation?

Arguments for Recovering Cooperation

- Conclusions that some have drawn from this analysis:
    - the game theory notion of rational action is wrong!
    - somehow the dilemma is being formulated wrongly
- Arguments to recover cooperation:
    - We are not all machiavelli!
    - The other prisoner is my twin!
    - Program equilibria and mediators
    - The shadow of the future…

The Iterated Prisoner's Dilemma

- One answer: play the game more than once.
- If you know you will be meeting your opponent again, then perhaps the incentive to defect evaporates. . . ?

- But… suppose you both know that you will play the game exactly n times.
- What should you do? Imagine yourself playing the final round.
- On round n, you have an incentive to defect, to gain that extra bit of payoff…
- But this makes round $n-1$ the last “real” round…but you have an incentive to defect there, too.
- This analysis technique is known as backwards induction.

Theorem

Playing the iterated Prisoner's Dilemma with a fixed, finite, pre-determined, commonly known number of rounds, mutual defection at every step is a dominant strategy equilibrium.

Infinitely Repeated Games

- Suppose you play the game an infinite number of rounds?
- Two issues:
  - How to measure utility over infinite plays?
    Summing utilities doesn't work – sums to infinity.
  - How to model strategies for infinite plays?
    Strategies are not just "C" or "D"

Utility functions for infinite runs

- Common approach: use a discount factor, $0 < \delta \leq 1$, to discount the value of future rounds – gives a finite value to infinite sum
- The value of the infinite run

$$\omega_0 \quad \omega_1 \quad \omega_2 \quad \omega_3 \quad \cdots \quad \omega_k \cdots$$

  to player i is then

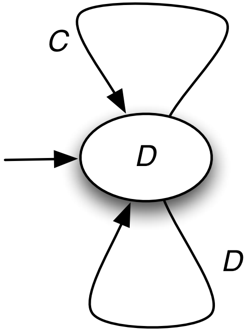
$$\sum_{k \in \mathbb{N}} \delta^k u_i(\omega_k)$$

- Alternative: compute average over all rounds.
  If players use automata strategies this is easy!

Strategies for infinite plays

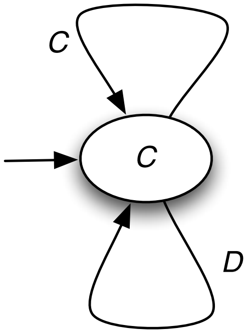
Strategies as automata strategies

- We represent strategies as finite automata – technically, Moore machines (“transducers”)
- Here is an automaton strategy called “ALLD”, which always defects:



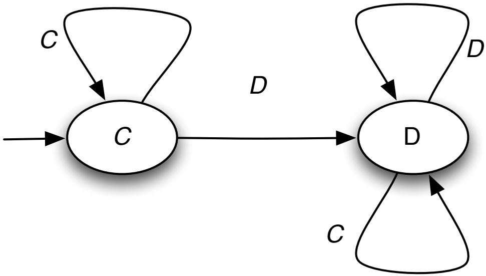
- Value inside a state is the action selected; outgoing arrows are actions of counterpart.

The ALLC strategy



Simply cooperates forever.

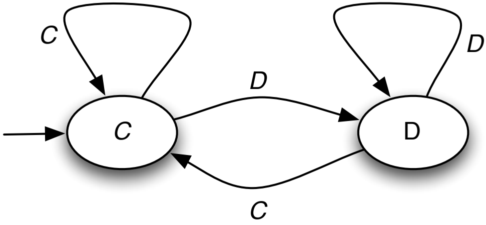
The GRIM strategy



I cooperate until you defect, at which point I flip to punishment mode: I defect forever after.

The TIT-FOR-TAT strategy

What does this strategy do?



Automata strategies playing against each other

Theorem

Finite machine strategies playing against each other will eventually enter a finite repeating sequence of outcomes.

The average utility of an infinite run is then simply the average utility over that finite repeating sequence.

ALLC against ALLC

round:	0	1	2	3	4	$\cdots$	
ALLC:	C	C	C	C	C	$\cdots$	average utility = -1
ALLC:	C	C	C	C	C	$\cdots$	average utility = -1

This is not a NE: either player would do better to choose another strategy (e.g., ALLD)

ALLC against ALLD

round:	0	1	2	3	4	$\cdots$	
ALLC:	C	C	C	C	C	$\cdots$	average utility = -3
ALLD:	D	D	D	D	D	$\cdots$	average utility = 0

This is not a NE: ALLC would do better to choose another strategy (e.g., ALLD)

ALLD against ALLD

round:	0	1	2	3	4	$\cdots$	
ALLD:	D	D	D	D	D	$\cdots$	average utility = -2
ALLD:	D	D	D	D	D	$\cdots$	average utility = -2

This is a NE (basically same as in one-shot case).
But it is not very desirable!

GRIM against ALLD

round:	0	1	2	3	4	$\cdots$	
GRIM:	C	D	D	D	D	$\cdots$	average utility = -2
ALLD:	D	D	D	D	D	$\cdots$	average utility = -2

Notice that GRIM tries to cooperate but then goes into punishment mode: on average, it doesn't do worse than if it had been ALLD.
This is not a NE: ALLD can beneficially deviate, as next slide shows.

GRIM against GRIM

round:	0	1	2	3	4	$\cdots$	
GRIM:	C	C	C	C	C	$\cdots$	average utility = -1
GRIM:	C	C	C	C	C	$\cdots$	average utility = -1

This <u>is</u> a NE! Rationally sustained cooperation.

The threat of punishment keeps players in line.

Nash Folk Theorem

In a game G, let player i's reservation value be the best utility that it can guarantee for itself, no matter what the other players do (i.e., even if they “gang up on it”).

Theorem (Nash Folk Theorem)

In an infinitely repeated game, every outcome in which every player gets at least their reservation value can be sustained as a Nash equilibrium.

In the infinitely repeated Prisoner's Dilemma, this means mutual cooperation can be sustained as an equilibrium.

Proof: use GRIM strategies. If any player deviates from required profile, other players punish him, ensuring he gets his reservation value.

- The strategy you really want to play in the prisoner's dilemma is:

  I'll cooperate if he will.

- Program equilibria[2] provide one way of enabling this.
- Each agent submits a program strategy to a mediator which jointly executes the strategies.

  Crucially, strategies can be conditioned on the strategies of the others.

[2] M. Tennenholtz, Program equilibrium, In *Games & Economic Behaviour*, 49(2), 1994.

Program Equilibria

- Consider the following program:

```
IF HisProgram == ThisProgram THEN
    DO(C);
ELSE
    DO(D);
END-IF.
```

- “==” is string comparison: comparing program texts.
- (Compare this with GRIM in iterated games.)
- The best response to this program is to submit the same program, giving an outcome of (C, C)!
- This is a program equilibrium.

A Folk Theorem for Program Equilibria

Theorem (Tennenholtz)

In any one shot game, every outcome in which every player gets at least their reservation value can be obtained as the outcome of a program equilibrium.

For the Prisoner's Dilemma, this means mutual cooperation can be obtained as the outcome of a program equilibrium.

Evolutionarily Games

Axelrod's Tournament

- Suppose you play iterated prisoner's dilemma against a range of opponents . . .
  What strategy should you choose, so as to maximise your overall payoff?
- Axelrod (1984) investigated this problem, with a computer tournament for programs playing the prisoner's dilemma[3].

[3] R. Axelrod, *The Evolution of Cooperation*, Basic Books, 1984.

Some strategies from Axelrod's Tournament

- ALLD:
  "Always defect" — the hawk strategy;
- TIT-FOR-TAT:
  1. On round $u = 0$, cooperate.
  2. On round $u > 0$, do what your opponent did on round $u - 1$.
- TESTER:
  On 1st round, defect. If the opponent retaliated, then play TIT-FOR-TAT. Otherwise intersperse cooperation & defection.
- JOSS:
  As TIT-FOR-TAT, except periodically defect.

Of the 63 strategies entered, he found TIT-FOR-TAT did best.

Recipes for Success in Axelrod's Tournament

Axelrod suggests the following rules for succeeding in his tournament:

- Don't be envious:
  Don't play as if it were zero sum!
- Be nice:
  Start by cooperating, and reciprocate cooperation.
- Retaliate appropriately:
  Always punish defection immediately, but use "measured" force — don't overdo it.
- Don't hold grudges:
  Always reciprocate cooperation immediately.

Note that TIT-FOR-TAT does well because it gets to play against other cooperative strategies: the "strategy population" consisted of other cooperative strategies.

Axelrod's evolutionary tournament

- Axelrod then suggested interpreting performance in his tournament as a measure of evolutionary fitness, and repeated the tournament over hundreds of generations.
- Strategies with higher relative fitness increased their presence in the strategy population compared to others.
- Notice that how well a strategy does depends on what other strategies are present in the population.
- Just assuming evolutionary forces, what will a population of strategies evolve to?
- Again, TIT-FOR-TAT did very well.

Evolutionary dynamics in Axelrod's Tournament

"The first thing that happens is that the lowest-ranking eleven entries fall to half their initial size by the fifth generation while the middle-ranking entries tend to hold their own and the top-ranking entries gradually grow in size. By the fiftieth generation, the [strategies] that ranked in the bottom third of the tournament have virtually disappeared, while most of those the middle third have started to shrink, and those in the top third are continuing to grow. The process simulates survival of the fittest. A [strategy] that is successful on average with the current distribution of [strategies] in the population will become an even larger proportion of the environment ... in the next generation. At first, a rule that is successful with all sorts of rules will proliferate, but later as the unsuccessful rules disappear, success requires success with other successful rules." (Axelrod 1984)

A solution concept for evolutionary games

- For Axelrod, the exciting thing was that TIT-FOR-TAT, and mutually sustained cooperation, could arise merely through blind evolutionary processes: cooperation through evolution.
- A strategy σ is an evolutionary stable strategy (ESS) if it will resist invasion by other strategies, assuming the population is made up initially of σ[4]

[4] J. Maynard Smith, *Evolution and the Theory of Games*, Cambridge UP, 1981.

Evolutionarily stable strategies (ESS)

Formally, strategy σ is an ESS iff:

1. It is a best response to itself.
   (Otherwise, other strategies could “prey” on it.)
2. For any strategy σ' that does as well against σ as σ does, σ does better against σ' than σ' does against itself.
   (So other strategies can’t benefit against σ by playing against themselves.)

(TIT-FOR-TAT is not, in fact, an ESS.)

Another Game

The Game of Chicken

		1	
		defect	coop
2	defect	1	2
		1	4
	coop	4	3
		2	3

- Think of James Dean in Rebel without a Cause: swerving = coop, driving straight = defect.
- Difference to prisoner's dilemma:

  Mutual defection is most feared outcome.

  (Whereas sucker's payoff is most feared in prisoner's dilemma.)

Solution Concepts

- There is no dominant strategy.
- Strategy pairs (C, D) and (D, C) are pure NE.
- All outcomes except (D, D) are Pareto optimal.
- All outcomes except (D, D) maximise social welfare.
- An anti-coordination game: players should choose different strategies.

Yet Another Game

A Coordination Game

		1: defect	1: coop
2	defect	1 (player 1) 1 (player 2)	0 (player 1) 0 (player 2)
	coop	0 (player 1) 0 (player 2)	1 (player 1) 1 (player 2)

- Here (C, C) and (D, D) are pure NE, but how do the players independently choose which to select?
- A coordination game, because the problem faced by players is how to coordinate.

Solving Coordination Games

1. Focal points:
   Sometimes outcomes in games have features that make them stand out, independently of the utility structure in games[5].
   Example: Suppose we are visiting Paris for a day, and get separated. Where do we meet up? In terms of utility, any place would do, but likely to pick a “landmark” $\rightarrow$ Eiffel Tower.
2. Evolutionary approaches:
   If we have time, we learn to coordinate (cf. ESS).

[5] T. C. Schelling, *The Strategy of Conflict*, Harvard UP, 1960

Other Symmetric 2×2 Games

- Given the 4 possible outcomes of (symmetric) cooperate/defect games, there are 24 possible orderings on outcomes.
  - $CC \succ_i CD \succ_i DC \succ_i DD$
    Cooperation dominates.
  - $DC \succ_i DD \succ_i CC \succ_i CD$
    Deadlock. You will always do best by defecting.
  - $DC \succ_i CC \succ_i DD \succ_i CD$
    Prisoner's dilemma.
  - $DC \succ_i CC \succ_i CD \succ_i DD$
    Chicken.
  - $CC \succ_i DC \succ_i DD \succ_i CD$
    Stag hunt.

Extensive Form Games

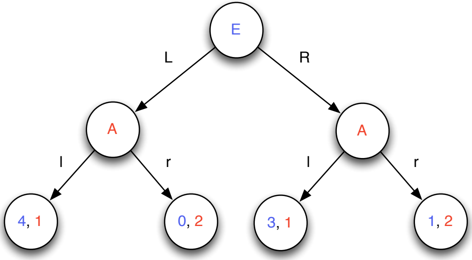
- Normal form games abstract away much of the detail in the structure of games, which usually involve taking moves in turn, leading to an ultimate payoff for both players when the game is over.
- Extensive form games explicitly capture this structure.
- In this lecture, we restrict ourselves to extensive form games with perfect information (everybody knows exactly what moves have been made previously) and no chance moves.

Game Trees

Extensive form games are usually modelled as game trees.

- A finite tree structure T
- The leaves of T are end games, and are labelled with payoffs for each player.
- Interior nodes of T are labelled with the player who makes a move at that point.
- Each edge leaving an interior node corresponds to a move that can be made by that player.
- The player at the root of the tree moves first.
- A strategy for player i is a function that selects a move for every interior node labelled with i.

Example Game Tree



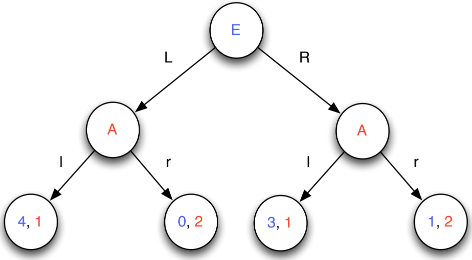
Two players: $N = \{E, A\}$.
First player to move is E; he can perform either L or R moves.

- Use backward induction to label every node with payoff profile that would be achieved in equilibrium (dynamic programming).
- Repeat the following:
  - If all the sub-children of an interior node have been labelled with a payoff profile, then label that node with a payoff profile from the children that maximises the payoff of the player making a turn at that node.
    (If there is a choice here, choose arbitrarily.)

  until all interior nodes have been labelled with payoff profiles.

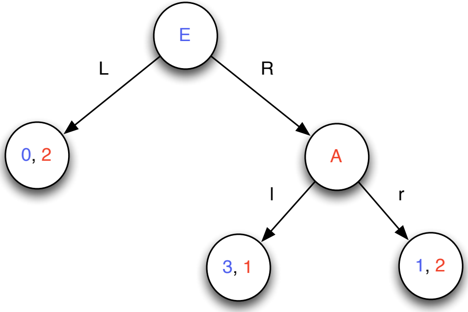
Illustrating Zermelo's Algorithm

Recall our game:



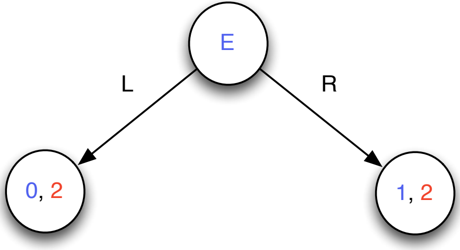
To illustrate the algorithm, we delete parts of the game tree that we have already "processed".
Initially, start with *A*'s bottom left choice: given a choice between 1 and 2, he will choose 2, i.e., move "r".

Illustrating Zermelo's Algorithm



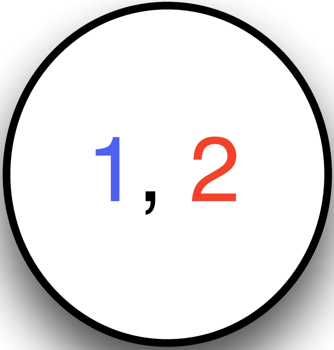
Now consider *A*'s bottom right choice: given a choice between 1 and 2, he will choose 2, i.e., move "r".

Illustrating Zermelo's Algorithm



Now consider *E*'s choice: he has a choice between 0 and 1 so will choose 1.

Illustrating Zermelo's Algorithm



So, player *E* receives 1 in equilibrium, while player *A* receives 2.

Properties of Zermelo's Algorithm

Theorem

Zermelo's algorithm terminates, leaving the root labelled with a payoff profile that would be obtained by a NE strategy profile.

The set of all such labelings is the set of all NE payoff profiles.

The algorithm runs in time polynomial in the size of the game tree.

Properties of extensive form games

Theorem

Every extensive form game (with perfect information and no chance moves) has a NE in pure strategies.

Proof: Zermelo's algorithm.

Theorem

Solving extensive form games is P-complete.

Zermelo's Algorithm in Computer Science

- One of the most phenomenally useful algorithms in computer science.
- Classic example of dynamic programming.
- Same algorithm is used in:
  - CTL model checking[6]
  - Computing optimal policies in Markov decision processes via "value iteration"[7]

[6] E. M. Clarke, O. Grumberg, and D. Peled. *Model Checking*, MIT Press, 1999. pages 35–39.

[7] M. L. Puterman, *Markov Decision Processes*, Wiley, 1994. pages 158–164.

Computational considerations

- Issues of representation:
  In a game with n players, where each player has m strategies, there are m^n possible outcomes: how do we represent utility functions $u_i(\cdots)$ in this case?
- Complexity issues:
  NE, PO, etc involve quantifying over strategies.
  Checking whether a game has a pure NE is NP-hard, even under very restrictive assumptions[8]
  Checking whether a game has a mixed NE is PPAD-complete[9]

---

[8] G. Gottlob, G. Greco, F. Scarcello. Pure Nash Equilibria: Hard and Easy Games. In *JAIR* 24:357–406, 2005.

[9] C. Daskalakis, P. W. Goldberg and C. H. Papadimitriou. The Complexity of Computing a Nash Equilibrium. In *SIAM Journal on Computing* 39(1):195-259, 2009.

An example compact non-cooperative game

Boolean Games

A Boolean game consists of:

- $N = \{1, \ldots, n\}$
  (the players)
- $\Phi = \{p, q, \ldots\}$
  (a finite set of Boolean variables)
- Φ_i
  (the set of variables under the control of i)
  The assignments that i can make to Φ_i are the actions available to i.
- γ_i
  (goal of agent i – the specification for i – propositional logic formula over Φ)

Outcomes

- A choice for agent i is an assignment

$$v_i : \Phi_i \to \mathbb{B}$$

  Agent i chooses a value for all its variables.
- An outcome is a collection of choices, one for each agent:

$$(v_1, \ldots, v_n)$$

Utility and Nash Equilibrium

The utility of outcome $(v_1, \ldots, v_n)$ to player i is:

$$u_i(v_1, \ldots, v_n) = \begin{cases} 1 & \text{if } (v_1, \ldots, v_n) \models \gamma_i \\ 0 & \text{otherwise.} \end{cases}$$

We can then define NE in the standard way.

An Example

Suppose:

$$\begin{array}{rcl} \Phi_1 & = & \{p\} \\ \Phi_2 & = & \{q, r\} \\ \gamma_1 & = & q \\ \gamma_2 & = & q \vee r \end{array}$$

Then $\gamma_1 \wedge \gamma_2$ is satisfied in NE.

Another Example
Matching pennies as a Boolean game

Suppose:

$$\begin{array}{rcl}\Phi_1 & = & \{p\} \\ \Phi_2 & = & \{q\} \\ \gamma_1 & = & p \leftrightarrow q \\ \gamma_2 & = & \neg(p \leftrightarrow q)\end{array}$$

There is no NE in this game.

Complexity of Boolean Games

Theorem

It is co-NP-complete to check whether an outcome forms a NE in a Boolean game.

It is Σ_2^p-complete to check whether a Boolean game has a NE.

Part IV

Cooperative games

Assumptions in Non-Cooperative Games

- Cooperation can't occur in the prisoner's dilemma because the conditions required for cooperation are not present, in particular:

  binding agreements are not possible.

- But suppose we drop this assumptions?
- In the real world, the role of contracts is to enable binding agreements.

Coalitional Games

- Coalitional games model scenarios where:
  - agents can benefit by cooperating;
  - binding agreements are possible.

Phases of Cooperative Action

- Issues in coalitional games (Sandholm et al, 1999):
  - Coalition structure generation.
  - Teamwork.
  - Dividing the benefits of cooperation.

Coalition Structure Generation

- Deciding in principle who will work together.
- The basic question:

  Which coalition should I join?

- The result: partitions agents into disjoint coalitions.
  The overall partition is a coalition structure.

Solving the optimization problem of each coalition

- Deciding how to work together.
- Solving the “joint problem” of a coalition.
- Finding how to maximise the utility of the coalition itself.
- Typically involves joint planning etc.

Dividing the Benefits

- Deciding "who gets what" in the payoff.
- Coalition members cannot ignore each other's preferences, because members can defect: if you try to give me a bad payoff, I can always walk away.
- We might want to consider issues such as fairness of the distribution.

Coalitional Games

A coalitional game is a pair:

$$G = \langle N, \nu \rangle$$

where:

- $N = \{1, \ldots, n\}$ is a set of players;
- $\nu : 2^N \to \mathbb{R}$ is the characteristic function of the game.

Usual interpretation: if $\nu(C) = k$, then coalition C can cooperate in such a way they will obtain utility k, which may then be distributed amongst team members.

Which Coalition Should I Join?

- Most important question in coalitional games:

  is a coalition stable?

  that is,

  is it rational for all members of coalition to stay with the coalition, or could they benefit by defecting from it?

- (There is no point in me trying to join a coalition with you unless you want to form one with me, and vice versa.)
- Stability is a necessary but not sufficient condition for coalitions to form.

Outcomes

- The core of a coalitional game is the set of feasible distributions of payoff to members of a coalition that no sub-coalition can reasonably object to.
- An outcome for a game $\langle N, \nu \rangle$ is a vector of payoffs to members of N, $\mathbf{x} = \langle x_1, \ldots, x_n \rangle$ which represents a feasible distribution of payoff to members of N.
  "Feasible" means:

$$\nu(N) \geq \sum_{i \in N} x_i$$

- Example: if $\nu(\{1, 2\}) = 20$, then possible outcomes are $\langle 20, 0 \rangle$, $\langle 19, 1 \rangle$, $\langle 18, 2 \rangle$, ..., $\langle 0, 20 \rangle$.
  (Actually there will be infinitely many!)
- An imputation is an outcome in which everybody is paid at least what they could earn themselves (individual rationality).

Objections

- Intuitively, a coalition C objects to an outcome if there is some outcome for them that makes all of them strictly better off.
- Where $\mathbf{x}$ is an outcome and $C \subseteq N$, we let

$$x_C = \sum_{i \in C} x_i$$

- Formally, $C \subseteq N$ objects to an outcome $\mathbf{x} = \langle x_1, \ldots, x_n \rangle$ iff $\nu(C) > x_C$.
- An outcome is not going to happen if a coalition objects to it, because this coalition could do better by defecting.

The Core

- The core, $\mathcal{C}$ of a game $G = \langle N, \nu \rangle$ is the set of outcomes to which no coalition objects:

$$\mathcal{C}(\langle N, \nu \rangle) = \{\mathbf{x} \mid \forall C \subseteq N : x_C \geq \nu(C)\}.$$

- If the core is non-empty then the grand coalition is stable, since nobody can benefit from defection.
- Thus, asking

  is the grand coalition stable?

  is the same as asking:

  is the core non-empty?

A game with an empty core

- Let $G = \langle N, \nu \rangle$ be a game with $G = \{1, 2, 3\}$ and

$$\nu(C) = \begin{cases} 1 & \text{if } |C| \geq 2 \\ 0 & \text{otherwise} \end{cases}$$

- Then $\mathcal{C}(G) = \emptyset$: any two players will always be able to deviate and share some "surplus".

Problems with the Core

- Sometimes, the core is empty; what happens then?
- Sometimes it is non-empty but isn't "fair".
  Suppose $N = \{1, 2\}$, $\nu(\{1\}) = 5$, $\nu(\{2\}) = 5$,
  $\nu(\{1, 2\}) = 20$.
  Then outcome $\langle 20, 0 \rangle$ (i.e., agent 1 gets everything) is not in the core, since the coalition $\{2\}$ can object. (He can work on his own and do better.)
  However, outcome $\langle 15, 5 \rangle$ is in the core: even though this seems unfair to agent 2, this agent has no objection.
- Why unfair? Because the agents are identical!

Fair Distribution: The Shapley Value

- The Shapley value is best known attempt to define how to divide benefits of cooperation fairly.
- Payments based on how much an agent contributes.
- The Shapley value $sh_i(G)$ of agent i in game G is the average amount that i is expected to contribute to a coalition.
- Axiomatically: a value which satisfies axioms:

  efficiency, symmetry, dummy player, and additivity.

Shapley's Axioms: (1) Efficiency

- The Shapley value $sh(G)$ is an imputation, i.e., a tuple of payoffs $sh(G) = (sh_1(G), \ldots, sh_n(G))$.
- It defines how the payoff of the grand coalition is distributed (standard assumption: superadditive games)
- The efficiency axiom says that no utility is wasted: all utility is distributed:

$$\nu(N) = \sum_{i \in N} sh_i(G).$$

Shapley's Axioms: (2) Symmetry

- The symmetry axiom says that agents which make the same contribution should get the same payoff.
- Let $\delta_i(S)$ be the amount that i adds by joining $S \subseteq N$:

$$\delta_i(S) = \nu(S \cup \{i\}) - \nu(S)$$

  ... the marginal contribution of i to S.
- Then i and j are interchangeable if $\delta_i(C) = \delta_j(C)$ for every $C \subseteq N \setminus \{i, j\}$.
- The symmetry axiom: if i and j are interchangeable then $sh_i(G) = sh_j(G)$.

Shapley's Axioms: (3) Dummy Player

- The dummy player axiom says that agents which make no contribution should get nothing.
- Formally, $i \in N$ is a dummy if $\delta_i(S) = 0$ for every $S \subseteq N \setminus \{i\}$.
- The dummy player axiom: if i is a dummy player then $\varphi_i = 0$.
- (Note: if i is a dummy then $\nu(\{i\}) = 0$ since $\delta_i(\emptyset) = 0$).

Shapley's Axioms: (4) Additivity

- This one is a bit technical! It basically says that if you combine two games, then the value a player gets should be the sum of the values in the individual games.
  You can't gain or lose by playing more than once.
- Formally, where $G^1 = (N, \nu^1)$ and $G^2 = (N, \nu^2)$ are games, define game $G^1 + G^2 = (N, \nu^1 + \nu^2)$ with:

$$\nu^1 + \nu^2(C) = \nu^1(C) + \nu^2(C).$$

- Then addititivy says:

$$sh_i(G^1 + G^2) = sh_i(G^1) + sh_i(G^2).$$

Shapley Defined

The Shapley value for i, denoted sh_i, is:

$$sh_i = \frac{1}{n!} \sum_{r \in R} \delta_i(S_i(r))$$

where R is the set of all orderings of N and $S_i(r)$ is the set of agents preceding i in ordering r.

Theorem (Shapley)

The Shapley value satisfies axioms (1)–(4) and, moreover, it is the only solution to these axioms.

Computational Issues in Coalitional Games

- It is important for an agent to know (eg) whether the core of a coalition is non-empty ...
  so, how hard is it to decide this?
- Problem: naive, obvious representation of coalitional game is exponential in the size of N!
- Now such a representation is:
  - utterly infeasible in practice; and
  - so large that it renders comparisons to this input size meaningless: stating that we have an algorithm that runs in (say) time linear in the size of such a representation means it runs in time exponential in the size of N!

How to Represent Characteristic Functions?

Two approaches to this problem:

- try to find a complete representation that is succinct in "most" cases
- try to find a representation that is not complete but is always succinct
- A common approach:
  interpret characteristic function over combinatorial structure.

What are Weighted Voting Games?

- A simple coalitional game, where value of any coalition is either 0 (“losing”) or 1 (“winning”).
- A type of yes/no voting system, in which a proposal (e.g., new law) is pitted against the status quo.
- For each agent $i \in N$, assign a weight w_i, and define an overall quota, q.

$$\nu(C) = \begin{cases} 1 & \text{if } \sum_{i \in C} w_i \geq q \\ 0 & \text{otherwise.} \end{cases}$$

Simple Majority Voting

- Weighted voting games are widely used in real world: for example simple majority voting is a special case.
- Simple majority voting is the political decision making system in, e.g., the UK and many other countries.
- Players are the people voting, e.g., the politicians deciding whether to pass a new law.
  Each player has weight $w_i = 1$ and for the threshold we have

  $$q = \lceil \frac{n+1}{2} \rceil.$$

  where n is the number of players (the size of the electorate).

Complexity of Weighted Voting Games

- Shapley value:
  #P-complete to compute, and “cannot be approximated”
  (Deng & Papadimitriou, 94) proved #P-completeness, and the other result follows from their construction.
- Core non-emptiness:
  in polynomial time.
  Core is non-empty iff there is an agent present in every winning coalition.

A Complete Representation

- Weighted voting games are not a complete representation for simple games:
  Some simple games cannot be represented using weighted voting games.
- k-weighted voting games are a complete representation for simple games, based on weighted voting games.

k-Weighted Voting Games

A k-weighted voting game S is a tuple

$$S = \langle N, \mathbf{w}_1, \ldots, \mathbf{w}_n, \mathbf{q} \rangle$$

where

- N is the set of voters
- $\mathbf{w}_i \in \mathbb{R}^k$ is a vector of k real weights for voter $i \in N$, and
- $\mathbf{q} \in \mathbb{R}^k$ is a vector of k real quotas.

k-Weighted Voting Games

- The coalitional characteristic function $\nu(C)$ is defined by

$$\nu(C) = \begin{cases} 1 & \text{if } \sum_{i \in C} \mathbf{w}_i \geq \mathbf{q} \\ 0 & \text{otherwise.} \end{cases}$$

  i.e., C wins in S if it wins in each of the component weighted voting games.
- So weighted voting games are a conjunction, and we often write $S = W_1 \wedge \cdots \wedge W_k$ to denote the k-weighted voting game composed from weighted voting games $W_1, \ldots, W_k$.

The EU: A 3-Weighted Voting Games

Voting in the EU is a 3-weighted voting game $W_1 \wedge W_2 \wedge W_3$, where the three weighted voting games corresponding to votes, countries, and population.

Each member state is a player.

The players are: {Germany, United Kingdom, France, Italy, Spain, Poland, Romania, The Netherlands, Greece, Czech Republic, Belgium, Hungary, Portugal, Sweden, Bu lgaria, Austria, Slovak Republic, Denmark, Finland, Ireland, Lithuania, Latvia, Slovenia, Estonia, Cyprus, Louxembourg, Malta}.

The EU: A 3-Weighted Voting Games

Voting in the EU is a 3-weighted voting game $W_1 \wedge W_2 \wedge W_3$, where the three weighted voting games corresponding to votes, countries, and population:

$$
\begin{aligned}
v_1 &= \{255; 29, 29, 29, 29, 27, 27, 14, 13, 12, 12, 12, 12, 12, 10, 10, 10, 7, 7, 7, 7, 7, \\
&\quad 4, 4, 4, 4, 4, 3\} \\
v_2 &= \{14; 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1\} \\
v_3 &= \{620; 170, 123, 122, 120, 82, 80, 47, 33, 22, 21, 21, 21, 21, 18, 17, 17, 11, \\
&\quad 11, 11, 8, 8, 5, 4, 3, 2, 1, 1\}
\end{aligned}
$$

Dimensionality

- k-weighted voting games are a complete representation for simple games: but how large does k have to be?
  The smallest number of components required is the dimension of the game.
- There exist simple coalitional games in which the smallest equivalent k weighted voting game is of dimension $\Omega(2^{n-1})$.
  Example: winning coalitions are of odd size!
- But: every simple coalitional game has an equivalent k weighted voting game of dimension $O(|2^N|)$.

Dimensionality Problems

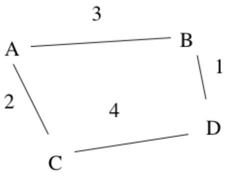
- We typically want the smallest representation possible.
- It is co-NP-complete to check whether two k-weighted voting games are equivalent, even if all weights are 0 or 1.
- It is NP-complete to check whether a given component of a k-weighted voting game is relevant.
- It is NP-complete to check whether a k-weighted voting game is minimal.

The Induced Subgraph Representation

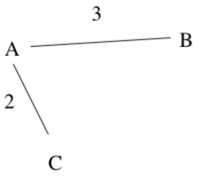
- Represent ν as an undirected graph on N, with integer weights $w_{i,j}$ between nodes $i, j \in N$.
- Value of coalition C then:

$$\nu(C) = \sum_{\{i,j\} \subseteq N} w_{i,j}$$

i.e., the value of a coalition $C \subseteq N$ is the weight of the subgraph induced by C.



the original graph defining v



subgraph induced by {A,B,C}
giving v({A,B,C}) = 3 + 2 = 5

Complexity of Induced Subgraphs

(Deng & Papadimitriou, 94)

- Computing Shapley: in polynomial time.
  An agent gets half the income from its edges.

$$\varphi_i = \frac{1}{2}\sum_{j \neq i} w_{i,j}$$

- However, determining emptiness of the core is NP-complete
- Checking whether a specific distribution is in the core is co-NP-complete

This representation is not complete.

Marginal Contribution Nets

(Ieong & Shoham, 2005)

- Characteristic function represented as rules:

$$\text{pattern} \longrightarrow \text{value.}$$

- Pattern is conjunction of agents, a rule applies to a group of agents C if C is a superset of the agents in the pattern. Value of a coalition is then sum over the values of all the rules that apply to the coalition.
  Example:

$$\begin{array}{rcl} a \wedge b & \longrightarrow & 5 \\ b & \longrightarrow & 2 \end{array}$$

  We have: $\nu(\{a\}) = 0$, $\nu(\{b\}) = 2$, and $\nu(\{a, b\}) = 7$.
- We can also allow negations in rules (agent not present).

Marginal Contribution Nets

- Shapley value: in polynomial time.
  Consider case where rules only contain positive literals, let ρ be set of such rules representing a game. Then:

$$\varphi_i = \sum_{r \in \rho : i \text{ in lhs of r}} \varphi_i^r$$

  where

$$\varphi_i^{\chi \longrightarrow x} = \frac{x}{|\chi|}$$

- Checking whether distribution is in the core is co-NP-complete
- Checking whether the core is non-empty is co-NP-hard.

A complete representation, but not necessarily succinct.

Qualitative Coalitional Games

- Often not interested in utilities, but in goals – either the goal is satisfied or not
- QCGs are a type of coalitional game in which each agent has a set of goals, and wants one of them to be achieved (doesn't care which)
  Agents cooperate in QCGs to achieve mutually satisfying sets of goals.
  Coalitions have sets of choices representing the different ways they could cooperate
  Each choice is a set of goals.

QCGs

A Qualitative Coalitional Game (QCG) is a structure:

$$\Gamma = \langle G, N, G_1, \ldots, G_n, V \rangle$$

where

- $G = \{g_1, \ldots, g_m\}$ is a set of possible goals;
- $N = \{1, \ldots, n\}$ is a set of agents;
- $G_i \subseteq G$ is a set of goals for each agent $i \in N$, the intended interpretation being that any of G_i would satisfy i;
- $V : 2^N \to 2^{2^G}$ is a characteristic function, which for every coalition $C \subseteq N$ determines a set $V(C)$ of choices, the intended interpretation being that if $G' \in V(C)$, then one of the choices available to coalition C is to bring about all the goals in G' simultaneously.

Feasible/Satisfying Goal Sets

- Goal set $G' \subseteq G$ satisfies an agent i if $G_i \cap G' \neq \emptyset$.
  Goal set $G' \subseteq G$ satisfies a coalition $C \subseteq N$ if

$$\forall i \in C, G_i \cap G' \neq \emptyset$$

- A goal set G' is feasible for C if $G' \in V(C)$.

Representing QCGs

- So, how do we represent the function $V : 2^N \to 2^{2^G}$?
- We use a formula Ψ_V of propositional logic over propositional variables N, G, such that:

$$\Psi[C, G'] = \top \text{ if and only if } G' \in V(C)$$

- "Often" permits succinct representations of V.
- Note that given Ψ_V, C, G', determining whether $G' \in V(C)$ can be done in time polynomial in size of C, G', Ψ_V.

Coalitional Resource Games (CRGs)

- Problem:

  where does characteristic function come from?

- One answer provided by Coalitional Resource Games (CRGs).
- Key ideas:
  - achieving a goal requires expenditure of resources;
  - each agent endowed with a profile of resources;
  - coalitions form to pool resource so as to achieve mutually satisfactory set of goals.

CRGs

A coalitional resource game Γ is an $(n+5)$-tuple:

$$\Gamma = \langle N, G, R, G_1, \ldots, G_n, \mathbf{en}, \mathbf{req} \rangle$$

where:

- $N = \{a_1, \ldots, a_n\}$ is a set of agents;
- $G = \{g_1, \ldots, g_m\}$ is a set of possible goals;
- $R = \{r_1, \ldots, r_t\}$ is a set of resources;
- for each $i \in N$, $G_i \subseteq G$ is a set of goals, as in QCGs;
- $\mathbf{en} : N \times R \to \mathbb{N}$ is an endowment function,
- $\mathbf{req} : G \times R \to \mathbb{N}$ is a requirement function.

Part V

Further reading

- Game Theory by Michael Maschler, Eilon Solan, and Shmuel Zamir. Cambridge UP, 2013.
  (IMHO, the best contemporary reference for game theory: rigorous but very readable.)
- A Course in Game Theory by Martin J. Osborne and Ariel Rubinstein. MIT Press, 1994.
  (Until Maschler *et al* came along, this was my favourite. Available free (legally!) from: `http://tinyurl.com/gtbook`)
- Game Theory – A Very Short Introduction by Ken Binmore. Oxford UP, 2007.
  (A useful companion for bedtime reading. Full of razor sharp opinions and insight from a master of the art.)

- Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations, by Yoav Shoham and Kevin Leyton-Brown. Cambridge UP, 2009.
  (A rigorous introduction to multi-agent systems as seen from a game theoretic perspective. Available free (legally!) from `http://www.masfoundations.org/mas.pdf`)
- Computational Aspects of Cooperative Game Theory by Georgios Chalkiadakis, Edith Elkind, and Michael Wooldridge. Morgan-Claypool, 2011.
  (As the name suggests, studies cooperative game theory from the point of view of computer science.)
- Algorithmic Game Theory. V. Vazirani, N. Nisan, T. Roughgarden, E. Tardos (eds). Cambridge UP, 2007.
  (Theoretical computer science take on GT/CS.)

Part VI

History of game theory

History of Game Theory

Phase One: 1928–54

- Originated in current form in early part of 20th century.
- Original focus: parlor games such as poker, chess (e.g., Zermelo on game of chess)
- First milestone: the minimax theorem proved in 1928 by Hungarian polymath John von Neumann (1903–57), leading to…
- Publication in 1944 of Theory of Games and Economic Behaviour by John von Neumann and Oskar Morgenstern (1902–77).
- Initial scope of game theoretic techniques very limited (typically “2 person zero sum games”)

- Scope of game theory hugely extended in 1950s with work of John Forbes Nash, Jr (1928–), and the concept of Nash equilibrium (NE)
  (NE remains to this day the chief analytical concept in game theory)
- A flurry of activity in 1950s, with other key results by Selten, Aumann, Shapley, Harsanyi and others
- But activity began to peter out as limitations to applicability of NE make themselves felt.

History of Game Theory

Phase Three: 1980–present

- In late 1970s/early 1980s, focus shifted to how societies converge on strategies.
- John Maynard Smith (1920–2004) and George Price (1922–75) laid foundations of evolutionary game theory, which refines NE and shows how societies can converge on equilibria through purely evolutionary processes
- Explain many biological questions, but also turn out to have direct relevance to economics.
- Robert Axelrod (1943–) hosts Prisoner's Dilemma competition, to much acclaim.

History of Game Theory

Phase Three: 1990–present

- Auction design raises much interest in game theoretic mechanism design
- Links between computer science & game theory: Christos Papadimitriou et al
- Four Nobel prizes for game theory:
  - 1994: John Harsanyi, John Forbes Nash, Reinhard Selten
  - 2005: Robert Aumann, Thomas Schelling
  - 2007: Leonid Hurwicz, Eric Maskin, Roger Myerson
  - 2012: Al Roth, Lloyd Shapley