Computational Game Theory

Lecture 1: Introduction & Motivation

UNIVERSITY OF OXFORD
DEPARTMENT OF COMPUTER SCIENCE

Part I

Orientation

What is Game Theory?

- The mathematical theory of

  ☞ **interaction between self-interested agents** ☜

- Self-interest: Agents ("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.
- Players must consider how each other player will act in order to make an optimal choice: hence **strategic** considerations.

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 games such as chess
  - such games are useful for highlighting key concepts
  - but the term “game” conveys something trivial :-(

When is Game Theory (In)Appropriate?

- When there are multiple loci of control and self interest
- If a system has a single designer/owner, then game theoretic analysis is probably inappropriate.
  - in this case the designer's problem is an **optimisation problem**
- If all players have same preferences, the challenge is **coordination**: ensuring that all players "pull in the same direction".

An Example Game

The Brexit Game

- The UK must choose whether to negotiate **hard** or **soft.**
- But the best choice depends on whether the EU choose hard or soft. . .
- The worst outcome for both parties is that both parties choose a hardline stance.
- Otherwise, best to choose the **opposite** of what your counterpart does:

☞ **a game of chicken** ☜

An Example Game

Issues or Insults?

- Trump and Clinton meet in a presidential debate
- They must each choose between **debating issues** or **making insults**
- What should Clinton do...?
- How well she is perceived to do will depend in part on the choice Trump makes
- What are the possible outcomes here? How do the candidates rank them?

Solution Concepts

What is a rational choice in a strategic setting?

- Key concern in game theory is to understand what the **outcomes** of a game will be, **under the assumption that players act rationally** (in pursuit of their preferences, given their beliefs).
- 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

Evaluating Solution Concepts

Existence: Does a solution concept **guarantee that the game has a rational outcome**?

Uniqueness: Does a solution concept guarantee that the rational outcome is **unique**?

Tractability: Is it computationally easy to verify that an outcome is rational? Is it easy to compute a rational outcome?

Comprehensibility: Is it easy for people to understand why an outcome is/is not rational?

Invariance: Do small changes in game setup lead to small changed in the outcome?

☞ **All solution concepts fail w.r.t. at least one.** ☜

Interpreting Game Theory

Descriptive Interpretations

- 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)

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 providing us **advice** about what we **ought** to do.
- Whether the advice is useful depends on whether the game model used was appropriate, and whether the **assumptions** on which the model depends are justified.
  (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**).
  - 3G spectrum auctions in 2000 yielded $35 billion for UK government.
  - "security games" paradigm (Milind Tambe)

Non-Cooperative *versus* Cooperative Games

- Game theory is divided into **non-cooperative** and **cooperative** versions.
- **Non-cooperative game theory** is concerned with settings where **players must act alone.** Solution concepts in non-cooperative game theory relate to **individual** choices.
- **Cooperative game theory** assumes players can make **binding agreements** to work together, allowing for teamwork, cooperation, joint action.
  - what teams (“coalitions”) will form?
  - how will teams share the cooperative dividend?

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 computational game theory, we have **protocols** + **preferences** + **rational choice**
- Take into account the fact that protocol participants are not benevolent entities – they are self-interested.

  strategic considerations come to the fore.
- Ignoring self interest in distributed systems misses a big part of the story.
  Example: sniping on eBay
- In **multi-agent systems**, mechanism participants are **software agents**.

Computer science for game theory:

How do we represent games? How computationally complex are solution concepts? How do we compute them efficiently?

Game theory for computer science:

How can game theory inform the design & analysis of computer 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 f for a class of games Γ with outcomes Ω is a function:

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

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

Computational issues for game theory

Non-emptiness: Given $G \in \Gamma$, is it the case that $f(G) \neq \emptyset$?
Does the game have any rational outcome according to the solution concept f?

Membership: Given $G \in \Gamma$ and $\omega \in \Omega_G$, is it the case that $\omega \in f(G)$?
Is a given outcome rational according to f?

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

Part III

Further reading

Further Reading

General Game Theory References

- **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 IV

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”)

History of Game Theory

Phase Two: 1954–1980

- Scope of game theory **hugely** extended in 1950s with work of John Forbes Nash, Jr (1928–2014), 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

Computational Game Theory

Lecture 2: Preferences, Utilities, and Decisions

UNIVERSITY OF OXFORD

DEPARTMENT OF COMPUTER SCIENCE

Overview

- Preferences are what give games their strategic character.
- In multi-agent systems, we **delegate** our preferences to a **software agent**, who then acts on our behalf in pursuit of them.
- Problem: people find it hard to formulate their preferences, and may not act rationally wrt their claimed preferences.
- **Preference elicitation** is the process of extracting preferences from **principals**
- **Utilities** are a **numeric representation of preferences**: allow us to **reduce decision-making to calculation**

Outcomes

- $\Omega = \{\omega_1, \ldots, \omega_k\}$ is the set of **outcomes**.
  In this course we usually assume Ω is **finite**.
- 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.

Certainty and Uncertainty

- We use slightly different interpretations of preferences, depending on whether the decision-making setting is one of **certainty** or **uncertainty**
- In **decision-making under certainty**, we know exactly what the consequences of our choices will be.
- In **decision-making under uncertainty**, we don't know exactly what the consequences of our choices will be: for every possible choice, there are **multiple** possible consequences, each with an attached probability.

Part V

Decision Making Under Certainty

Capturing Preferences

- The idea is that given any two outcomes ω, ω', we are able to say which is our most preferred
- In other words, we can **rank** outcomes
- Rankings are not required to be strict — you are allowed to be **indifferent** between outcomes
- Formally, we capture such rankings in **preference relations**

Preference Relations

A **preference relation** is a binary relation $\succeq \; \subseteq \Omega \times \Omega$, which is required to satisfy:

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

Indifference

If both

$$\omega \succeq \omega' \quad \text{and} \quad \omega' \succeq \omega$$

then we say

you are indifferent between ω and ω'

and we write

$$\omega \sim \omega'$$

Strict Preference

If

$$\omega \succeq \omega' \quad \text{but not} \quad \omega' \succeq \omega$$

then we say

you strictly prefer ω over ω'

and we write

$$\omega \succ \omega'$$

Interpreting Preferences (IMPORTANT)

Revealed Preferences

- Preferences are not observable, but **choices** are.

☞ **revealed preferences** ☜

- $\omega \succ \omega'$ means that:
  - if you have a choice between ω and ω', **you will choose** ω
  - if you have a choice between two options, one of which will result in ω, the other of which will result in ω', you will choose the option resulting in ω

Interpreting Preferences (IMPORTANT)

Rational Choice Doesn’t Make You Scrooge

- A preference relation **must encompass everything that would influence decision making**.
- If you care about other people, then **this is reflected in your preferences**.
- Preference theory and rational choice theory cope equally well with angels and devils.
- 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 indicate more preferred.
- The numbers are called **utility values**, **utilities**, or **payoffs**.
- A **utility function** $u : \Omega \to \mathbb{R}$ is said to represent a preference relation $\succeq$ iff we have:

$$u(\omega) \geq u(\omega') \qquad \text{iff} \qquad \omega \succeq \omega'$$

Theorem

For every preference relation $\succeq\ \subseteq \Omega \times \Omega$ *there is a utility function* $u : \Omega \to \mathbb{R}$ *that represents* $\succeq$*.*

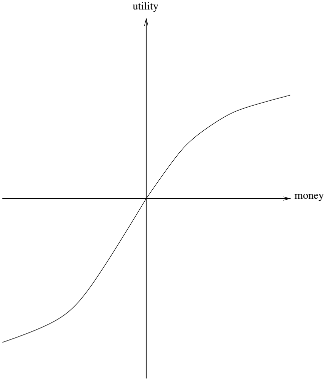
Proof: exercise.

What is Utility?

- We use numeric utilities to represent preferences because it reduces rational choice to a calculation.
- Utilities are **selected** to represent preferences $\succeq$.
- It is a **fallacy** to claim you choose ω over ω' because $u(\omega) > u(\omega')$.
  - You make this choice because $\omega \succ \omega'$.
  - The $u(\cdots)$ values were chosen to reflect this.
- But, if we picked the numbers right, then you **behave as though** you were maximising utility.
- Utility values in decision-making under certainty don't represent **intensity**: they are **ordinal** values, which indicate **relative** rankings.
- **Interpersonal comparisons of utility are difficult.** There is no celcius scale for utility! "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.
- Typical relationship between utility & money:



(Don't take the graph too literally…)

Outcome Functions

- Let Σ be the set of **strategies** (choices, actions, alternatives. . .) available to our decision maker.
- An **outcome function** (a.k.a. consequences function) is

$$g : \Sigma \to \Omega$$

- The **feasible outcomes** are those that could be obtained through the performance of an appropriate strategy. Formally, the feasible outcomes are the range of g, i.e., ran g.
- If ran $g \subset \Omega$ then some outcomes are not feasible.

Decision Making Under Certainty

- A problem of **decision making under certainty** is given by a quad
$$\langle \Omega, \quad u : \Omega \to \mathbb{R}, \quad \Sigma, \quad g : \Sigma \to \Omega \rangle$$
- The task of our decision maker is to select a strategy σ^* that leads to an outcome which maximises utility:
$$\sigma^* \in \arg\max_{\sigma \in \Sigma} u(g(\sigma))$$
- This is an **optimisation problem.**

Part VI

Decision Making Under Uncertainty

Motivation

- In most settings, we don't know exactly what outcome will result by performing a strategy.
- For every strategy $\sigma \in \Sigma$, there will typically be a range of possible outcomes, with differing probabilities of occurring.
- Such settings require more complex machinery for preferences and utilities.
- In particular, preference relations $\succeq \subseteq \Omega \times \Omega$ are not enough: we need preferences over **lotteries**.

Reminder: Probability Distributions

- A **probability distribution over a non-empty set** S is a function

  $$f : S \rightarrow [0, 1]$$

  which must satisfy the constraint that

  $$\sum_{s \in S} f(s) = 1$$

- So $f(s)$ is the probability of s given distribution f
- Let $\Delta(S)$ denote the set of all probability distributions over S
- Where $s \in S$ and $f \in \Delta(S)$, we sometimes write $P(s, f)$ to denote $f(s)$

Lotteries

- A **lottery** over S is a probability distribution over S.
- We denote individual lotteries by ℓ, ℓ', ℓ_1, etc, and denote the set of lotteries over set S by $lott(S)$.

Example

Suppose $\Omega = \{whisky, gin, brandy\}$. Then

$$\ell_1 = \frac{1}{10}\text{ whisky} + \frac{2}{10}\text{ gin} + \frac{7}{10}\text{ brandy}$$

means whisky with probability 0.1, gin with probability 0.2, brandy with probability 0.7.

An Example

Example

Suppose

$$\ell_1 = \frac{1}{10} gin + \frac{9}{10} brandy \qquad \text{and} \qquad \ell_2 = brandy$$

Here, ℓ_2 is a **degenerate** lottery (brandy with certainty!)

If I prefer brandy over gin, what is an appropriate preference relation over these lotteries?

Suppose you prefer gin over brandy?

Compound Lotteries

A **compound lottery** is a lottery over lotteries – an element of the set $lott(lott(\Omega))$.

Example

Recall

$$\ell_1 = \frac{1}{10} gin + \frac{9}{10} brandy \qquad \text{and} \qquad \ell_2 = 1\, brandy$$

Now suppose:

$$\ell_3 = \frac{9}{10}\ell_1 + \frac{1}{10}\ell_2 \qquad \text{and} \qquad \ell_4 = \frac{1}{100}\ell_1 + \frac{99}{100}\ell_2$$

Compound Lotteries

- For each $u \in \mathbb{N}$, define the set $lott_u(\Omega)$ by

$$lott_u(\Omega) = \begin{cases} lott(\Omega) & \text{if } u = 0 \\ lott(lott_{u-1}(\Omega)) \cup lott_{u-1}(\Omega) & \text{otherwise.} \end{cases}$$

- Finally define $Lott(\Omega)$ by:

$$Lott(\Omega) = \bigcup_{u \in \mathbb{N}} lott_u(\Omega)$$

- Any compound lottery can be reduced to a simple lottery (an element of $lott_0(\Omega) = lott(\Omega)$) by multiplying out probabilities.

☞ **We work with preference relations over $Lott(\Omega)$** ☜

How do we Measure the Utility of a Lottery?

Expected Utility

- How can we use a utility function $u : \Omega \rightarrow \mathbb{R}$ to measure the utility of a **lottery**?
- We use **expected utility** – intuitively, the "average" utility that we could expect from the lottery.
- More precisely, the expected value of the function $u(\cdots)$.
- Given a utility function $u : \Omega \rightarrow \mathbb{R}$, the expected utility $EU(\ell)$ of lottery ℓ is defined by:

$$EU(\ell) = \sum_{\omega \in \Omega} u(\omega) P(\omega, \ell)$$

From Preferences to Utilities Under Uncertainty

- Suppose I give you a preference relation

  $$\succeq \; \subseteq Lott(\Omega) \times Lott(\Omega)$$

  Can you give me a utility function

  $$u : \Omega \rightarrow \mathbb{R}$$

  so that

  $$\ell_1 \succeq \ell_2 \text{ iff } EU(\ell_1) \geq EU(\ell_2)$$

- The answer is "yes" if the preference relation satisfies some additional properties, due to John von Neumann and Oskar Morgenstern
- A utility function $u : \Omega \rightarrow \mathbb{R}$ can represent preferences over lotteries **if and only if** $\succeq$ satsfies these axioms

Important: Utility in decision making under uncertainty must capture the intensity of preferences.

A Warm Up Exercise

Win-Lose Lotteries

- Suppose $\Omega = \{\mathcal{W}, \mathcal{L}\}$ with $\mathcal{W}$ = "win" and $\mathcal{L}$ = "lose", with $\mathcal{W} \succ \mathcal{L}$.
- Then we only need one additional axiom, **continuity**, which says that you prefer to maximise the probability of a win:

  For all lotteries ℓ_1, ℓ_2, we have

$$\ell_1 \succeq \ell_2 \quad \textit{iff} \quad P(\mathcal{W}, \ell_1) \geq P(\mathcal{W}, \ell_2)$$

A Warm Up Exercise

Win-Lose Lotteries

Theorem

A preference relation $\succeq \subseteq Lott(\{\mathcal{W}, \mathcal{L}\}) \times Lott(\{\mathcal{W}, \mathcal{L}\})$ *over win-lose lotteries satisfies completeness, reflexivity, transitivity, and continuity iff there exists a utility function*

$$u : \{\mathcal{W}, \mathcal{L}\} \to \mathbb{R}$$

such that

$$\ell_1 \succeq \ell_2 \quad \textit{iff} \quad EU(\ell_1) \geq EU(\ell_2)$$

where

$$EU(\ell) = \sum_{\omega \in \{\mathcal{W}, \mathcal{L}\}} u(\omega) P(\omega, \ell)$$

Proof: exercise.

Von Neumann and Morgenstern's Axioms

Now let's look at the general case.
In addition to **completeness**, **reflexivity**, and **transitivity**, Von Neumann and Morgenstern introduced:

1. the **Equivalence** axiom
2. the **Monotonicity** axiom
3. the **Archimedean** axiom
4. the **Independence/Substitution** axiom

Von Neumann and Morgenstern’s Axioms

The Equivalence Axiom

The **structure** of a lottery is irrelevant – all that matters is the probability distribution over outcomes that the lottery defines.

> *Every compound lottery is ranked in exactly the same way as the simple lottery with the same probability distribution over outcomes.*

If you prefer ℓ_1 over ℓ_2 then you will prefer to maximise the probability of getting ℓ_1 over ℓ_2.

Suppose

$$\ell_1 \succ \ell_2$$

Then

$$p \geq q$$

iff

$$p\ell_1 + (1 - p)\ell_2 \ \succeq \ q\ell_1 + (1 - q)\ell_2$$

Essentially, this says we can quantify our preferences over lotteries.

If

$$\ell_1 \succeq \ell_2 \succeq \ell_3$$

then there is some $p \in [0, 1]$ *such that*

$$\ell_2 \quad \sim \quad p\ell_1 + (1 - p)\ell_3$$

Von Neumann and Morgenstern's Axioms

The Independence Axiom (sometimes called "substitution")

We can **freely substitute** lotteries that we are indifferent between.
(In much the same we can freely substitute equal terms when manipulating algebraic expressions.)

Von Neumann and Morgenstern's Theorem

Theorem

A preference relation $\succeq \subseteq$ *Lott*$(\Omega) \times$ *Lott*(Ω) *satisfies the von Neumann and Morgenstern axioms iff there exists a function*

$$u : \Omega \rightarrow \mathbb{R}$$

such that:

$$\ell_1 \succeq \ell_2 \quad \textit{iff} \quad EU(\ell_1) \geq EU(\ell_2)$$

where

$$EU(\ell) = \sum_{\omega \in \Omega} u(\omega) P(\omega, \ell)$$

Von Neumann and Morgenstern's Theorem

Proof Overview

NB: we only prove the left $\rightarrow$ right direction.

1. Identify **best and worst outcomes** – call them $\mathcal{W}$ and $\mathcal{L}$
2. Use $\mathcal{W}$ and $\mathcal{L}$ to establish a **scale** with $\mathcal{L}$ valued 0 and $\mathcal{W}$ valued 1.
3. Use the Archimedean axiom to place outcomes ω on this scale.

Von Neumann and Morgenstern's Theorem

Proof Step 1: Dealing with the trivial case

- If
  $$\omega_1 \sim \omega_2 \sim \cdots \sim \omega_k$$
  then we are indifferent between all outcomes.
- In this case define $u(\omega) = 1$ for all $\omega \in \Omega$
- $\ldots$ and we are done.

Von Neumann and Morgenstern's Theorem

Proof Step 2: Establish a scale from worst to best

- Otherwise, order the alternatives worst up to best.
  (Assume for simplicity the ordering is strict: no indifference.)
- Such an ordering exists by the completeness requirement.
- Pick the lowest ranked outcome; call it $\mathcal{L}$ ("lose").
  Let $u(\mathcal{L}) = 0$.
- Pick the highest ranked outcome; call it $\mathcal{W}$ ("win").
  Let $u(\mathcal{W}) = 1$.
- $\mathcal{L}$ and $\mathcal{W}$ define our scale, within which we fit other outcomes.

Von Neumann and Morgenstern's Theorem

Proof Step 3: Ordering the alternatives

- Where $p \in [0, 1]$, we let $\ell^*(p)$ denote the following lottery:

$$\ell^*(p) = p\mathcal{W} + (1 - p)\mathcal{L}$$

- By the **Archimedean axiom**, for each outcome ω there is a $p_\omega \in [0, 1]$ such that $\omega \sim \ell^*(p_\omega)$, i.e.,

$$\omega \quad \sim \quad p_\omega\mathcal{W} + (1 - p_\omega)\mathcal{L}$$

- Define $u(\omega) = p_\omega$
- The probability p_ω places ω on the scale between $\mathcal{L}$ and $\mathcal{W}$

- Take two lotteries $\ell_1 \succeq \ell_2$, where

$$\begin{aligned} \ell_1 &= p_1\omega_1 + \cdots + p_k\omega_k \\ \ell_2 &= q_1\omega_1 + \cdots + q_k\omega_k \end{aligned}$$

- Replace each occurrence of ω_i in ℓ_1, ℓ_2 by $\ell^*(p_{\omega_i})$.
  So, for example, we have:

$$\ell_1 = p_1\ell^*(p_{\omega_1}) + \cdots + p_k\ell^*(p_{\omega_k})$$

- We now have ℓ_1 and ℓ_2 expressed in terms of $\mathcal{W}$ and $\mathcal{L}$

- Collect $\mathcal{W}$ and $\mathcal{L}$ terms and simplify. Looking at ℓ_1:

$$\begin{aligned} \ell_1 &= (\textstyle\sum_{i=1}^k p_i p_{\omega_i})\mathcal{W} + (1 - (\textstyle\sum_{i=1}^k p_i p_{\omega_i}))\mathcal{L} \\ &= (\textstyle\sum_{i=1}^k p_i u(\omega_i))\mathcal{W} + (1 - (\textstyle\sum_{i=1}^k p_i u(\omega_i)))\mathcal{L} \end{aligned}$$

  Similarly:

$$\ell_2 = (\textstyle\sum_{i=1}^k q_i u(\omega_i))\mathcal{W} + (1 - (\textstyle\sum_{i=1}^k q_i u(\omega_i)))\mathcal{L}$$

Von Neumann and Morgenstern's Theorem

Proof Step 4: Correctness of the construction (cont'd)

Since $\mathcal{W} \succ \mathcal{L}$, by monotonicity it must be that

$$\sum_{i=1}^{k} p_i u(\omega_i) \geq \sum_{i=1}^{k} q_i u(\omega_i)$$

Since $u(\mathcal{W}) = 1$ and $u(\mathcal{L}) = 0$, then

$$\begin{array}{rcl} EU(\ell_1) & = & \sum_{i=1}^{k} p_i u(\omega_i) \\ EU(\ell_2) & = & \sum_{i=1}^{k} q_i u(\omega_i) \end{array}$$

Therefore

$$EU(\ell_1) \geq EU(\ell_2)$$

Decision Making Under Uncertainty

A problem of **decision making under uncertainty** is given by a quad

$$\langle \Omega, \quad u : \Omega \to \mathbb{R}, \quad \Sigma, \quad g : \Sigma \to \mathit{Lott}(\Omega) \rangle$$

The task of our decision maker is to select a strategy σ^* that **maximises expected utility**:

$$\begin{aligned} \sigma^* &\in \arg\max_{\sigma \in \Sigma} EU(g(\sigma)) \\ &\in \arg\max_{\sigma \in \Sigma} \sum_{\omega \in \Omega} u(\omega) P(\omega, g(\sigma)) \end{aligned}$$

This is probably the most important single definition underpinning contemporary AI.

Part VII

Paradoxes of Expected Utility Theory

Paradoxes of Expected Utility Theory

- The MEU paradigm is so entrenched that it is commonplace to define **rational agents** as those that act so as to maximise expected utility.
- However, it is not hard to find examples in which either maximising expected utility seems to be the wrong thing to do, or where the advice offered by the theory is counter to strong intuitions.

Paradoxes of Expected Utility Theory

Paradox 1: A Simple Example

Example

Suppose

$$\ell_1 = \$50 \qquad \ell_2 = \frac{1}{2}\$101 + \frac{1}{2}\$0$$

Many people prefer ℓ_1, and the certain $\$50$.

An example of **risk aversity**, and an illustration that expected monetary reward does not equate to utility.

Example

Consider following lotteries:

$$\begin{aligned} \ell_A &= \$2m \\ \ell_B &= \tfrac{89}{100}\$2m + \tfrac{1}{10}\$10m + \tfrac{1}{100}\$0 \end{aligned}$$

[2] M. Allais. Le Comportement de l'Homme Rationnel devant le Risque: Critique des Postulats et Axiomes de l'Ecole Americaine, *Econometrica*, 21(4), 1953.

Paradoxes of Expected Utility Theory

Paradox 2: The Allais Paradox[2]

Example

Consider following lotteries:

$$\ell_A = \$2m$$

$$\ell_B = \frac{89}{100}\$2m + \frac{1}{10}\$10m + \frac{1}{100}\$0$$

Most people prefer ℓ_A over ℓ_B.

[2] M. Allais. Le Comportement de l'Homme Rationnel devant le Risque: Critique des Postulats et Axiomes de l'Ecole Americaine, *Econometrica*, 21(4), 1953.

Paradoxes of Expected Utility Theory

Paradox 2: The Allais Paradox[2]

Example

Consider following lotteries:

$$\ell_A = \$2m$$

$$\ell_B = \frac{89}{100}\$2m + \frac{1}{10}\$10m + \frac{1}{100}\$0$$

Most people prefer ℓ_A over ℓ_B.

$$\ell_C = \frac{11}{100}\$2m + \frac{89}{100}\$0$$

$$\ell_D = \frac{1}{10}\$10m + \frac{9}{10}\$0$$

[2] M. Allais. Le Comportement de l'Homme Rationnel devant le Risque: Critique des Postulats et Axiomes de l'Ecole Americaine, *Econometrica*, 21(4), 1953.

Paradoxes of Expected Utility Theory

Paradox 2: The Allais Paradox[2]

Example

Consider following lotteries:

$$\begin{aligned} \ell_A &= \$2m \\ \ell_B &= \tfrac{89}{100}\$2m + \tfrac{1}{10}\$10m + \tfrac{1}{100}\$0 \end{aligned}$$

Most people prefer ℓ_A over ℓ_B.

$$\begin{aligned} \ell_C &= \tfrac{11}{100}\$2m + \tfrac{89}{100}\$0 \\ \ell_D &= \tfrac{1}{10}\$10m + \tfrac{9}{10}\$0 \end{aligned}$$

Most people prefer ℓ_D over ℓ_C.

---

[2] M. Allais. Le Comportement de l'Homme Rationnel devant le Risque: Critique des Postulats et Axiomes de l'Ecole Americaine, *Econometrica*, 21(4), 1953.

Paradoxes of Expected Utility Theory

The Allais Paradox

Lemma

If you have preferences $\ell_A \succ \ell_B$ and $\ell_D \succ \ell_C$ then you do not satisfy the Von Neumann and Morgenstern axioms.

Proof: Let $u : \Omega \to \mathbb{R}$ represent $\succ$, and let $x = u(\$0)$, $y = u(\$2m)$, and $z = u(\$10m)$.

Lemma

If you have preferences $\ell_A \succ \ell_B$ and $\ell_D \succ \ell_C$ then you do not satisfy the Von Neumann and Morgenstern axioms.

Proof: Let $u : \Omega \to \mathbb{R}$ represent $\succ$, and let $x = u(\$0)$, $y = u(\$2m)$, and $z = u(\$10m)$.

$$\begin{aligned} EU(\ell_A) &> EU(\ell_B) && (1) \\ y &> 0.1z + 0.89y + 0.01x && (2) \end{aligned}$$

Lemma

If you have preferences $\ell_A \succ \ell_B$ and $\ell_D \succ \ell_C$ then you do not satisfy the Von Neumann and Morgenstern axioms.

Proof: Let $u : \Omega \to \mathbb{R}$ represent $\succ$, and let $x = u(\$0)$, $y = u(\$2m)$, and $z = u(\$10m)$.

$$EU(\ell_A) > EU(\ell_B) \tag{1}$$
$$y > 0.1z + 0.89y + 0.01x \tag{2}$$

$$EU(\ell_D) > EU(\ell_C) \tag{3}$$
$$0.1z + 0.9x > 0.11y + 0.89x \tag{4}$$

Lemma

If you have preferences $\ell_A \succ \ell_B$ and $\ell_D \succ \ell_C$ then you do not satisfy the Von Neumann and Morgenstern axioms.

Proof: Let $u : \Omega \to \mathbb{R}$ represent $\succ$, and let $x = u(\$0)$, $y = u(\$2m)$, and $z = u(\$10m)$.

$$
\begin{aligned}
EU(\ell_A) &> EU(\ell_B) && (1)\\
y &> 0.1z + 0.89y + 0.01x && (2)
\end{aligned}
$$

$$
\begin{aligned}
EU(\ell_D) &> EU(\ell_C) && (3)\\
0.1z + 0.9x &> 0.11y + 0.89x && (4)
\end{aligned}
$$

But add $0.89(x - y)$ to each side of (2):

$$
\begin{aligned}
0.11y + 0.89x &> 0.1z + 0.9x && (5)\\
EU(\ell_C) &> EU(\ell_D) && (6)
\end{aligned}
$$

Contradiction.

Paradoxes of Expected Utility Theory

Paradox 3: Kahneman and Tversky's Framing Effects[3]

Example

A deadly strain of flu has been detected in the USA, which is expected to kill 600 people. There are only 2 treatments:

(A) 200 people will be saved

(B) there is a $\frac{1}{3}$ probability that 600 will be saved, and a $\frac{2}{3}$ probability that nobody will be saved

[3] A. Tversky and D. Kahneman. The Framing of Decisions and the Psychology of Choice. *Science*, 221, 1981.

Paradoxes of Expected Utility Theory

Paradox 3: Kahneman and Tversky's Framing Effects[3]

Example

A deadly strain of flu has been detected in the USA, which is expected to kill 600 people. There are only 2 treatments:

(A) 200 people will be saved

(B) there is a $\frac{1}{3}$ probability that 600 will be saved, and a $\frac{2}{3}$ probability that nobody will be saved

72% of people had $A \succ B$.

[3] A. Tversky and D. Kahneman. The Framing of Decisions and the Psychology of Choice. *Science*, 221, 1981.

Example

A deadly strain of flu has been detected in the USA, which is expected to kill 600 people. There are only 2 treatments:

- **(A)** 200 people will be saved
- **(B)** there is a $\frac{1}{3}$ probability that 600 will be saved, and a $\frac{2}{3}$ probability that nobody will be saved

72% of people had $A \succ B$. Now consider the following.

- **(C)** 400 people will die
- **(D)** there is a $\frac{1}{3}$ probability that nobody will die, and a $\frac{2}{3}$ probability that 600 people will die

[3] A. Tversky and D. Kahneman. The Framing of Decisions and the Psychology of Choice. *Science*, 221, 1981.

Paradoxes of Expected Utility Theory

Paradox 3: Kahneman and Tversky's Framing Effects[3]

Example

A deadly strain of flu has been detected in the USA, which is expected to kill 600 people. There are only 2 treatments:

(A) 200 people will be saved

(B) there is a $\frac{1}{3}$ probability that 600 will be saved, and a $\frac{2}{3}$ probability that nobody will be saved

72% of people had $A \succ B$. Now consider the following.

(C) 400 people will die

(D) there is a $\frac{1}{3}$ probability that nobody will die, and a $\frac{2}{3}$ probability that 600 people will die

78% of people had $D \succ C$.

[3] A. Tversky and D. Kahneman. The Framing of Decisions and the Psychology of Choice. *Science*, 221, 1981.

Paradoxes of Expected Utility Theory

Paradox 3: Kahneman and Tversky's Framing Effects[3]

Example

A deadly strain of flu has been detected in the USA, which is expected to kill 600 people. There are only 2 treatments:

- **(A)** 200 people will be saved
- **(B)** there is a $\frac{1}{3}$ probability that 600 will be saved, and a $\frac{2}{3}$ probability that nobody will be saved

72% of people had $A \succ B$. Now consider the following.

- **(C)** 400 people will die
- **(D)** there is a $\frac{1}{3}$ probability that nobody will die, and a $\frac{2}{3}$ probability that 600 people will die

78% of people had $D \succ C$.
But A and C are identical, as are B and D...

[3] A. Tversky and D. Kahneman. The Framing of Decisions and the Psychology of Choice. *Science*, 221, 1981.

Paradoxes of Expected Utility Theory

Kahneman and Tversky's Framing Effects

- The issue here is that people are affected by the way a decision problem is "framed"
- In this example, we prefer to choose "saving lives"
- The study of **how people make economic decisions** is the domain of **behavioural economics**
- Fun reading on this subject:

  D. Ariely. *Predictably Irrational*. Harper-Collins, 2008.

Part VIII

Preference Relations with Structure

Preference Relations with Structure

- Preference relations often have some structure.
- Here we look at 3 important classes of preference relation:
  1. single-peaked preferences
  2. dichotomous preferences
  3. lexicographic preferences

Single-Peaked Preferences

A preference relation is **single-peaked** with respect to a fixed ordering $\omega_1 > \omega_2 > \ldots > \omega_k$ of the alternatives (the **axis**) iff

1. there is a most preferred candidate ω^* and;
2. candidates closer to ω^* are preferred over those that are further away:
   - if $\omega^* > \omega_1 > \omega_2$ then $\omega_1 \succ \omega_2$
   - if $\omega_1 > \omega_2 > \omega^*$ then $\omega_2 \succ \omega_1$

Example

Suppose we can order electoral candidates according to on the left-right spectrum. It is natural for us to identify a single point on this spectrum representing our personal political preferences, and we prefer candidates closer to this ideal.

Single-peaked preferences are important in social choice theory, where they play a key role in the **median voter theorem**.

Dichotomous Preferences

- A preference relation is **dichotomous** if it classifies all outcomes as either **win** or **lose**.
- Formally, there exist $\mathcal{W} \subseteq \Omega$ and $\mathcal{L} \subseteq \Omega$ such that:
  - $\mathcal{W} \cup \mathcal{L} = \Omega$
  - $\mathcal{W} \cap \mathcal{L} = \emptyset$
  - $\forall \omega_1, \omega_2 \in \mathcal{W}$, $\omega_1 \sim \omega_2$
  - $\forall \omega_1, \omega_2 \in \mathcal{L}$, $\omega_1 \sim \omega_2$
  - $\forall \omega_1 \in \mathcal{W}$, $\forall \omega_2 \in \mathcal{L}$, $\omega_1 \succ \omega_2$
- The set $\mathcal{W}$ can be interpreted as a **goal**.
- Dichotomous preferences naturally specified with logical formulae.

Lexicographic Preferences

Preferences are lexicographic if outcomes can be characterised by an ordered set of attributes, where each attribute has its own ordering.

Example

Let $\Omega =$ all words in the Oxford English dictionary.
Suppose I prefer words occurring earlier in the dictionary.
Hence I prefer all words starting with “a” over those starting with “b”, and all words starting “aa” over those starting “ab”, and so on.
$\Rightarrow$ why such preferences are called **lexicographic**.

Lexicographic Preferences

Example

With respect to cars, the attributes I use to order are:

$$\text{colour} > \text{engine} > \text{nationality}$$

The ordering for each of these attributes is:

1. **colour**: red $\succ$ blue $\succ$ green
2. **engine type**: electric $\succ$ petrol $\succ$ diesel
3. **nationality**: German $\succ$ French $\succ$ UK

So, I rank all red cars above all other colours.
I rank all red electric cars above all red petrol cars.
I rank all red electric German cars above all red electric UK cars.

Part IX

Compact Representations

The Need for Compact Representations

Often, the set Ω is too large to enumerate preference relations explicitly.

☞ **we need compact & tractable representations** ☜

Example

Suppose you are in a class with n other people, and you must form a team with some subset of them. Your preferences must order 2^n possible teams…

But compact representations raise **computational** problems: decision problems start to get hard!
Compact but tractable representations of utilities/preferences is a major area of research.

Boolean Domains

Many domains can be represented by a finite set of variables $\Phi = \{x_1, \ldots, x_l\}$, where each variable takes value $\top$ (“true”) or $\bot$ (false)

Example

Recall the class team example. Let $N = \{1, \ldots, n\}$ be the class members. For each class member $i \in N$ define a Boolean variable x_i, with

$x_i = \top$ means “i is in the team”

$x_i = \bot$ means “i is <u>not</u> in the team”

Any valuation $v : \Phi \to \{\top, \bot\}$ defines a team.

Dichotomous Boolean Preferences

Example

Continuing the team example. Suppose you have dichotomous preferences: you divide the teams into $\mathcal{W} \subseteq \mathbf{2}^N$ and $\mathcal{L} \subseteq \mathbf{2}^N$, such that $\mathcal{W} \cup \mathcal{L} = \mathbf{2}^N$ and $\mathcal{W} \cap \mathcal{L} = \emptyset$.

- We can specify dichotomous such preference relations via **propositional formulae**, γ.
- Each $i \in N$ corresponds to a Boolean variable x_i
- The set of **satisfying assignments** for γ are the “winning” teams.
- We can define a utility function:

$$u(v) = \begin{cases} 1 & \text{if } v \models \gamma \\ 0 & \text{otherwise} \end{cases}$$

Dichotomous Preferences

The Fab Four

Example

Suppose

$$\gamma = \mathit{John} \vee \mathit{Paul} \wedge (\mathit{George} \wedge \mathit{Ringo}) \wedge \neg(\mathit{John} \wedge \mathit{Paul})$$

Which teams satisfy this goal?

Let the **naive representation** for dichotomous Boolean preferences be the representation in which we **explicitly list all winning sets** $\mathcal{W} \subseteq \mathbf{2}^N$.

Theorem

1. *The propositional formula representation for dichotomous Boolean preferences is* ***complete****: any dichotomous Boolean preference relation can be represented by a propositional formula.*
2. *The propositional formula representation can be* ***exponentially more compact*** *than the naive representation.*
3. *There exist dichotomous preference relations for which the smallest propositional representation is of size exponential in* $|N|$ *(and hence no better than the naive representation).*

Weighted Formula Representations

- What about utility functions $u : \mathbf{2}^N \to \mathbb{R}$?
  The **naive representation** here involves listing all $2^{|N|}$ input/output pairs of u
- We can use **weighted formula representation.**
- A **weighted formula**, or **rule** is a pair (φ, x) where φ is a propositional formula and $x \in \mathbb{R}$.
  We sometimes write $\varphi \longrightarrow x$
- We use **rule bases**, $\mathcal{R}$, to define utility functions:

$$\mathcal{R} = \{(\varphi_1, x_1), \ldots, (\varphi_k, x_k)\}$$

- The utility function u associated with $\mathcal{R}$ is defined:

$$u_{\mathcal{R}}(v) = \sum_{\substack{(\varphi_i, x_i) \in \mathcal{R} \\ v \models \varphi_i}} x_i$$

Weighted Formula Representations

Theorem

1. *The weighted formula representation is a complete representation for utility functions* $u : \mathbf{2}^N \to \mathbb{R}$
2. *The weighted formula representation can be* ***exponentially more compact*** *than the naive representation.*
3. *There exist utility functions* $u : \mathbf{2}^N \to \mathbb{R}$ *for which the smallest weighted formula representation requires exponentially many rules.*

Weighted Formula Representations

Theorem

1. *Given a target value $k \in \mathbb{R}$ and rulebase $\mathcal{R}$, the problem of determining whether there exists a valuation v such that $u_{\mathcal{R}}(v) \geq k$ is NP-complete.*

2. *The problem of finding an **optimal** valuation v^* satisfying*

$$v^* \in \arg\max_{v} u_{\mathcal{R}}(v)$$

is FP^{NP}-complete.
(This means it is as hard as the travelling salesman problem: to solve it requires a polynomial number of queries to an NP oracle.)

We can use **binary search** to find an optimal valuation with queries to the NP-oracle.

Part X

An Application: Preferences for Combinatorial Auctions

Preferences for Combinatorial Auctions

- Auctions for bundles of goods.
- A good example of bundles of good are spectrum licences.
- For the 1.7 to 1.72 GHz band for Brooklyn to be useful, you need a license for Manhattan, Queens, Staten Island.
- Most valuable are the licenses for the same bandwidth.
- But a different bandwidth licence is more valuable than no license

Valuation Functions

- Let $\mathcal{Z} = \{z_1, \ldots, z_m\}$ be a set of items to be auctioned.
- We capture preferences of agent i with a **valuation** function:

$$u_i : \mathbf{2}^{\mathcal{Z}} \rightarrow \mathbb{R}$$

- Thus, for every possible bundle of goods $Z \subseteq \mathcal{Z}$, $u_i(Z)$ says how much Z is worth to i.

Properties of Valuation Functions

- If

$$u_i(\emptyset) = 0$$

  then we say that the valuation function for i is **normalised.**
- A common assumption is **free disposal**:

$$Z_1 \subseteq Z_2 \quad \text{implies} \quad u_i(Z_1) \leq u_i(Z_2)$$

- Free disposal means an agent is never worse off having more stuff.

Bids

- Rather than exhaustive evaluations, allow bidders to construct valuations from the bids they want to mention.
- **Atomic bids** take the form

  $$(Z, p)$$

  where
  - $Z \subseteq \mathcal{Z}$
  - $p \in \mathbb{R}_+$
- A bundle Z' **satisfies** a bid (Z, p) if $Z \subseteq Z'$.
- In other words a bundle satisifes a bid if it contains at least the things in the bid.

Atomic Bids

- Atomic bids define valuations

$$u_\beta(Z') = \begin{cases} p & \text{if } Z' \text{ satisfies } (Z, p) \\ 0 & \text{otherwise} \end{cases}$$

- Atomic bids alone don't allow us to construct very interesting valuations.

XOR Bids

- To construct more complex valuations, atomic bids can be combined into more complex bids.
- One approach is XOR bids

$$\beta_1 = (\{a, b\}, 3) \quad \text{XOR} \quad (\{c, d\}, 5)$$

- XOR because we will pay for **at most one.**
- We read the bid to mean:

  I would pay 3 for a bundle that contains a and b but not c and d. I will pay 5 for a bundle that contains c and d but not a and b, and I will pay 5 for a bundle that contains a, b, c and d.

- From this we can construct a valuation.

XOR Bids

The valuation function corresponding to

$$\beta_1 = (\{a, b\}, 3) \quad \text{XOR} \quad (\{c, d\}, 5)$$

is thus:

$$\begin{aligned}
u_{\beta_1}(\{a\}) &= 0 \\
u_{\beta_1}(\{b\}) &= 0 \\
u_{\beta_1}(\{a, b\}) &= 3 \\
u_{\beta_1}(\{c, d\}) &= 5 \\
u_{\beta_1}(\{a, b, c, d\}) &= 5
\end{aligned}$$

XOR Bids

More formally, the following XOR bid:

$$\beta = (Z_1, p_1) \text{ XOR } \cdots \text{ XOR } (Z_k, p_k)$$

defines a valuation u_β as follows:

$$u_\beta(Z') = \begin{cases} 0 & \text{if } Z' \text{ doesn't satisfy any } (Z_i, p_i) \\ \max\{p_i \mid Z_i \subseteq Z'\} & \text{otherwise} \end{cases}$$

- XOR bids are **fully expressive**, that is they can express any valuation function over a set of goods.
- To do that, we may need an exponentially large number of atomic bids.
- However, the valuation of a bundle can be computed in polynomial time.

Computational Game Theory

Lecture 3: Normal Form Games

UNIVERSITY OF OXFORD

DEPARTMENT OF COMPUTER SCIENCE

Part XI

Game Forms and Games

Strategic (Normal) Form Non-Cooperative Games

- In this lecture we study **strategic form non-cooperative games** and their solution concepts
- This is the best-known class of games
- Also called **normal form games**
- Recall that in **non-cooperative games**, players must act **alone –** joint decisions are not possible

Game Forms

- Let $N = \{1, \ldots, n\}$ be the set of **players.**
- Each player i must simultaneously chooses a **strategy** from their set of **pure strategies** Σ_i
- As a result of the **combination** of strategies selected, an outcome in Ω will result.
- Consequences of collective decisions captured by an **outcome function**:

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

- A **game form** is a then structure:

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

Adding preferences

- We now assume each player has preferences captured by a utility function $u_i(\cdots)$.
- It is useful to drop reference to outcomes Ω, and instead give utility functions over **combinations of choices**.
- Thus instead of

  $$u_i : \Omega \to \mathbb{R}$$

  we write

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

  with the understanding that

  $$u_i(\sigma_1, \ldots, \sigma_n)$$

  is shorthand for

  $$u_i(g(\sigma_1, \ldots, \sigma_n)).$$

- So, in what follows, 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 of **players**;
- Σ_i is a set of **(pure) strategies** for player $i \in N$;
- $u_i : \Sigma_1 \times \cdots \times \Sigma_n \to \mathbb{R}$ is the **utility function** for agent $i \in N$.

The utility i gets depends not on only **her** actions, but on the actions of **others**, and similarly for other agents.

Payoff Matrices

We can neatly capture a two player game in a **payoff matrix**:

1 \ 2	l	c	r
T	x_1, x_2	x_3, x_4	x_5, x_6
B	x_7, x_8	x_9, x_{10}	x_{11}, x_{12}

- Agent 1 is the **row player.** Strategies for this player correspond to rows: $\Sigma_1 = \{T, B\}$
- Agent 2 is the **column player.** Strategies for this player correspond to columns: $\Sigma_2 = \{l, c, r\}$
- Each cell lists utilities from the corresponding outcome
- Two player games sometimes called **bimatrix games**

A Running Example

$$
\begin{aligned}
&N = \{1, 2\}, \\
&\Sigma_1 = \{T, B\}, \quad \Sigma_2 = \{L, R\}, \\
&u_1(T, L) = 1, \quad u_1(T, R) = 1, \\
&u_1(B, L) = 0, \quad u_1(B, R) = 0 \\
&u_2(T, L) = 1, \quad u_2(T, R) = 0, \\
&u_2(B, L) = 0, \quad u_2(B, R) = 1.
\end{aligned}
$$

1 \ 2	L	R
T	1 1	0 1
B	0 0	1 0

A Running Example

$$
\begin{aligned}
&N = \{1, 2\}, \\
&\Sigma_1 = \{T, B\}, \quad \Sigma_2 = \{L, R\}, \\
&u_1(T, L) = 1, \quad u_1(T, R) = 1, \\
&u_1(B, L) = 0, \quad u_1(B, R) = 0 \\
&u_2(T, L) = 1, \quad u_2(T, R) = 0, \\
&u_2(B, L) = 0, \quad u_2(B, R) = 1.
\end{aligned}
$$

1 \ 2	L	R
T	1, 1	1, 0
B	0, 0	0, 1

If you were player 1, what would you do?

Part XII

Solution Concepts

Solution Concepts

- If players act rationally, what will the outcome of the game be?
- Answered by **solution concepts**
- Key solution concepts for strategic games:
  - dominant strategies
  - Nash equilibria
  - iterated elimination equilibrium
- **Best response** is a key concept to understand these.

Strategy Profiles

A **strategy profile**, $\vec{\sigma}$, is a tuple of strategies, one for each player:

$$\vec{\sigma} = (\sigma_1, \ldots, \sigma_i, \ldots, \sigma_n) \in \Sigma_1 \times \cdots \times \Sigma_i \times \cdots \times \Sigma_n$$

We denote the strategy profile obtained by replacing the i component of $\vec{\sigma}$ with σ'_i by

$$(\vec{\sigma}_{-i}, \sigma'_i)$$

And so:

$$(\vec{\sigma}_{-i}, \sigma'_i) = (\sigma_1, \ldots, \sigma'_i, \ldots, \sigma_n)$$

We sometimes refer to Σ_{-i}, with obvious interpretation

Dominant Strategies

- Suppose you have a strategy σ, with the following property:

  no matter what choice you made, my best response to that choice would be to choose σ*.*

- Strategies that have this property are called **dominant strategies.**
- The fact that a strategy is dominant is a pretty compelling argument for choosing it: it is **never** a sub-optimal decision.

Dominant Strategies

Formally, we say $\sigma_i \in \Sigma_i$ is a **dominant strategy** if:

for all $\vec{\sigma}$ and for all $\sigma_i' \in \Sigma_i$, we have

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

Note: Comp Sci books tend to use this definition, but are often vague. Traditional GT books use a slightly different definition. Unless otherwise noted, when we refer to "dominant strategies", this is the definition we intend.

Back to the Running Example

1 \ 2	L	R
T	1 1	0 1
B	0 0	1 0

- T is a dominant strategy for player 1.
- There is no dominant strategy for player 2:
  - if 1 plays T then best response is L
  - if 1 plays B then best response is R

Weakly Dominant Strategies

NB: Older books use this definition

We say $\sigma_i \in \Sigma_i$ is **a weakly dominant strategy** if:

1. for all $\vec{\sigma}$ and for all $\sigma_i' \in \Sigma_i$, we have

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

2. for some $\vec{\sigma}$ we have

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

   for all $\sigma_i' \in \Sigma_i$

With a weakly dominant strategy: you can never be worse off, and in at least one case will be strictly better off, than any other.

Strictly Dominant Strategies

We say $\sigma_i \in \Sigma_i$ is **a strictly dominant strategy** if

for all $\vec{\sigma}$ and for all $\sigma_i' \in \Sigma_i$, we have

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

With a strictly dominant strategy: you are always strictly better off than any other strategy.

Back to the Running Example

1 \ 2	L	R
T	1, 1	1, 0
B	0, 0	0, 1

- T is in fact a strictly dominant strategy for player 1.
- (Obviously) no strictly dominant strategy for 1.

Dominant Strategy Equilibrium

- A **dominant strategy equilibrium** is a strategy profile in which every player has chosen a dominant strategy.
- A very strong solution concept…but unfortunately, there isn't always a dominant strategy.

(Pure Strategy) Nash Equilibrium

- Dominant strategies are required to be best responses to **all** counterpart strategies. This is a very strong requirement!
- **Nash equilibrium** relaxes this requirement, and is therefore more widely applicable.
- 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

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

- Let $NE(G)$ denote the Nash equilibria of G.
- **Nobody can benefit by deviating from a Nash equilibrium.**

Back to the Running Example

1 \ 2	L	R
T	1, 1	1, 0
B	0, 0	0, 1

- (T,L) is the unique NE
- For every other outcome, at least one player has a beneficial deviation.
- For example, with (B,R), player 1 would benefit by deviating to T.

Problems with Nash Equilibrium

- Not every game has a (pure) NE.
- Some games have more than one NE

  ☞ **equilibrium selection problem** ☜

- Some NE are bad (have undesirable social properties)

Part XIII

The Concept of “Best Response”

Best Response

- A important way of understanding solution concepts is through the idea of a **best response** function.
- A player's best response to a strategy profile $\vec{\sigma}$ is the choice that would give that player highest utility, assuming the other players made choices as defined in $\vec{\sigma}$.

Best Responses

For each player i define a **best response function**:

$$BR_i : \Sigma_1 \times \cdots \times \Sigma_n \to \mathbf{2}^{\Sigma_i}$$

as follows:

$$BR_i(\vec{\sigma}) = \arg\max_{\sigma_i \in \Sigma_i} u_i(\vec{\sigma}_{-i}, \sigma_i)$$

$BR_i(\vec{\sigma})$ will be non-empty, but not guaranteed to be a singleton.

We define the **best response function of the game** as follows:

$$BR(\vec{\sigma}) = BR_1(\vec{\sigma}) \times \cdots \times BR_n(\vec{\sigma})$$

Back to the Running Example

1 \ 2	L	R
T	1, 1	1, 0
B	0, 0	0, 1

- $BR_1(L) = \{T\}$
- $BR_1(R) = \{T\}$
- $BR_2(T) = \{L\}$
- $BR_2(B) = \{R\}$

NE as a Fixed Point of the Best Response Function

- $s \in S$ is a **fixed point** of a function $f : S \to S$ if $s = f(s)$.
- $s \in S$ is a fixed point of a function $f : S \to \mathbf{2}^S$ if $s \in f(s)$.
- NE can naturally be characterised in terms of fixed points & best responses… a fact which turns out to be very important later…

Lemma

$$\vec{\sigma} \in NE(G) \qquad \textit{iff} \qquad \vec{\sigma} \in BR(\vec{\sigma})$$

In other words, the Nash equilibria of a game are precisely the fixed points of the game's best response function:

$$NE(G) = \{\vec{\sigma} \in \Sigma_1 \times \cdots \times \Sigma_n \mid \vec{\sigma} \in BR(\vec{\sigma})\}.$$

Dominant Strategies and Best Responses

Lemma

$\sigma_i \in \Sigma_i$ is a dominant strategy for player i iff

$$\sigma_i \in \bigcap_{\vec{\sigma} \in \Sigma} BR_i(\vec{\sigma})$$

Part XIV

Social Welfare

What would a Benevolent God choose?

- Suppose an omniscient, impartial, benevolent external entity was able to **choose the outcome of the game.**
- What would they choose?
- Intuitively, the outcome that is **best for the society**
- This is the realm of **social welfare**
- The answer is not obvious, because **interpersonal comparisons of utility are very difficult**
- Key notions:
  - Pareto optimality
  - utilitarian social welfare
  - egalitarian social welfare

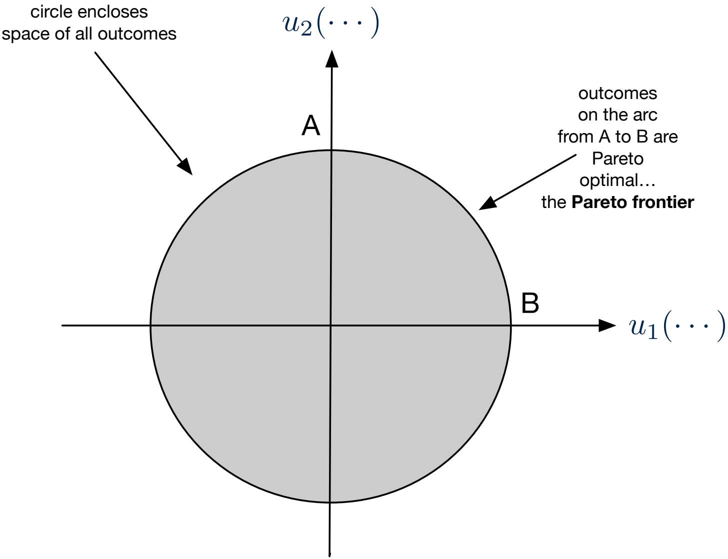
Pareto Optimality

(Also called Pareto Efficiency)

- A strategy profile is **Pareto optimal** (a.k.a. **Pareto efficient**) if there is no other outcome that makes one agent **better off** without making another agent **worse off**.
- If $\vec{\sigma}$ **is** Pareto optimal, then at least one agent will be reluctant to move away from it (because this agent will be worse off).
- If $\vec{\sigma}$ **is not** Pareto optimal, then:
  - $\vec{\sigma}$ is **inefficient**: it is "wasting" utility
  - we can make a **Pareto improvement** which nobody would object to.
- Pareto optimality is probably the least contentious notion of social welfare.

Graphical Illustration of Pareto Optimality

Each strategy profile $\vec{\sigma}$ gives a coordinate $(u_1(\vec{\sigma}), u_2(\vec{\sigma}))$.



Utilitarian Social Welfare

- The utilitarian social welfare of $\vec{\sigma}$ is the **sum of utilities** that each agent gets from $\vec{\sigma}$
- An outcome $\vec{\sigma}^*$ that maximises utilitarian social welfare thus satisfies:

$$\vec{\sigma}^* \in \arg\max_{\vec{\sigma}} \sum_{i \in N} u_i(\vec{\sigma})$$

- Intuitively the "total amount of wealth that $\vec{\sigma}$ creates".
- Problems:
  - it doesn't look at the **distribution** of utility
  - **utilities are not on same scale** (like adding temperatures in celcius and fahrenheit. . .)
- Appropriate when the whole system (all agents) has a single owner (then overall benefit of the system is important, not individuals).

Egalitarian Social Welfare

- Egalitarian social welfare says that we should **try to make the worst off member of society as well off as possible**
- A strategy profile $\vec{\sigma}^*$ that maximises egalitarian social welfare will satisfy:

$$\vec{\sigma}^* \in \arg\max_{\vec{\sigma}'} \min\{u_i(\vec{\sigma}') \mid i \in N\}$$

- Intuitive justification: John Rawls' **veil of ignorance**[4]:

  Suppose you could choose a society, but without any knowledge of where you would be placed in that society. Then, Rawls argued, you would choose the society that maximises egalitarian social welfare.

[4] J. Rawls. *A Theory of Justice*. Belknap Press, 1971.

Back to the Running Example

1 \ 2	L	R
T	1 1	0 1
B	0 0	1 0

(T,L) maximises utilitarian and egalitarian social welfare, and is the only Pareto efficient outcome.

Relationships Between Solution Concepts

Theorem

1. *Every dominant strategy equilibrium is a Nash equilibrium, but the converse need not be the case.*
2. *Nash equilibria and dominant strategy need not be Pareto efficient, nor need they maximise utilitarian/egalitarian social welfare.*
3. *Any outcome that maximises utilitarian social welfare is Pareto efficient, but the converse need not be the case.*

Part XV

Some Important Games

The Prisoner's Dilemma

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

They both know 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;*
- *if neither confesses, then they will each be jailed for one year.*

*The prisoners **only** care about minimising the amount of time they spend in prison.*"

Confession is **defection** (*D*)

Keeping quiet is **cooperation** (*C*)

Payoff matrix for the Prisoner's Dilemma

1 \ 2	D	C
D	-2, -2	0, -3
C	-3, 0	-1, -1

Payoff matrix for the Prisoner's Dilemma

1 \ 2	D	C
D	-2, -2	0, -3
C	-3, 0	-1, -1

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

Dominant Strategy Analysis

- Consider player 1's analysis:

Dominant Strategy Analysis

- Consider player 1's analysis:
  - Suppose 2 defects: my best response is to defect.

Dominant Strategy Analysis

- Consider player 1's analysis:
  - Suppose 2 defects: my best response is to defect.
  - Suppose 2 cooperates: my best response is to defect.

Dominant Strategy Analysis

- Consider player 1's analysis:
  - Suppose 2 defects: my best response is to defect.
  - Suppose 2 cooperates: my best response is to defect.
  - Defection is a best response to **all** of 2's actions.

Dominant Strategy Analysis

- Consider player 1's analysis:
  - Suppose 2 defects: my best response is to defect.
  - Suppose 2 cooperates: my best response is to defect.
  - Defection is a best response to **all** of 2's actions.
  - $\Rightarrow$ defection is a dominant strategy for 1.
- The game is **symmetric**: defection is also a dominant strategy for player 2.

  ☞ (D, D) **is a dominant strategy equilibrium** ☜

  ...in which both serve two years in jail
- But **intuition** says this is **not** the best outcome:

  Surely they should both cooperate – then they each serve just one 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!

- The apparent paradox has made the game famous
- 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!
  - the dilemma is being formulated wrongly!
- Arguments to recover cooperation:
  - × We are not all machiavelli!
  - × The other prisoner is my twin!
  - ✓ Program equilibria
  - ✓ The shadow of the future...

☞ **Cooperation doesn't occur in the PD because the conditions required for cooperation are not present** ☜

The Game of Chicken

1 \ 2	D	C
D	1, 1	4, 2
C	2, 4	3, 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

- No dominant strategy.
- (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.

A Coordination Game

How to choose between multiple similar equilibria?

1 \ 2	L	R
T	1, 1	0, 0
B	0, 0	1, 1

- Here (T, L) and (B, R) 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.

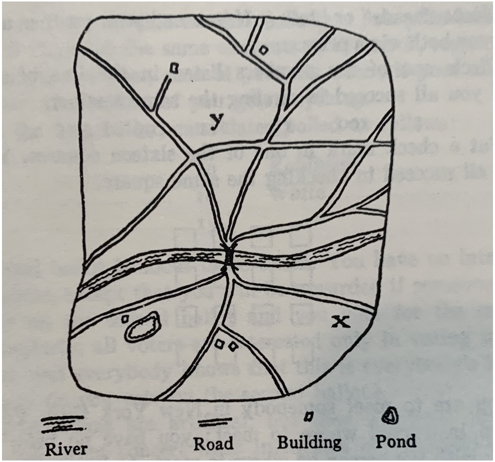
Solving Coordination Games

1. **Focal points**:
   Sometimes outcomes in games have features that make them stand out, independently of the utility structure in games[5].
2. **Evolutionary approaches**:
   If we have time, we **learn** to coordinate (cf. ESS).

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

Focal Points

From T. C. Schelling, *The Strategy of Conflict*, page 55.



Suppose you and I got separated in this area with no way of communicating and no prior arrangement. **Where should we head for?**

The Stag Hunt

If it was a matter of hunting a deer, everyone well realised that he must remain faithful to his post; but if a hare happened to pass within reach of one of them, we cannot doubt that he would have gone off in puruit of it without scruple.”

Rousseau (A Discourse on Inequality)

The Stag Hunt

1 \ 2	Deer	Hare
Deer	7, 7	0, 4
Hare	4, 0	4, 4

- Another social dilemma, but less painful than the prisoner’s dilemma.
- Here there are **two** pure NE.

The Hawk-Dove Game

In the Hawk-Dove Game, individuals meet to obtain a particular resource (e.g., food) from the environment.
Hawks are fierce; Doves are timid...

1. When a Hawk competes with a Dove, the Hawk takes the whole of the resource.
2. When a Dove competes with a Dove, they share the resource equally.
3. When a Hawk competes with a Hawk, they fight; they end up sharing the resource but incur the cost of fighting.

This game plays an important role in **evolutionary game theory**.

The Hawk-Dove Game

V denotes the **value of the resource.**
C denotes the **cost of injury.**

	Hawk	Dove
Hawk	$\frac{V}{2} - C$ $\frac{V}{2} - C$	0 V
Dove	V 0	$\frac{V}{2}$ $\frac{V}{2}$

Solution concepts depend on the relations between V and C.

NE in the Hawk-Dove Game

	Hawk	Dove
Hawk	$\frac{V}{2} - C$ $\frac{V}{2} - C$	0 V
Dove	V 0	$\frac{V}{2}$ $\frac{V}{2}$

- If $V/2 > C$:

  $$(H, H)$$

- If $C > V/2$:

  $$(D, H), (H, D)$$

- If $C = V/2$:

  $$(H, H), (D, H), (H, D)$$

Matching Pennies

A Game with No Pure Nash Equilibrium

- Players 1 and 2 each have a $1 coin
- They simultaneously show one face of their coin (either “heads” or “tails”)
- If they show the same face, then 1 takes 2’s coin.
- If they show different faces, then 2 takes 1’s coin.

Matching Pennies Payoff Matrix

1 \ 2	Heads	Tails
Heads	-1 1	1 -1
Tails	1 -1	-1 1

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.

Part XVI

Eliminating Dominated Strategies

Dominated Strategies

Given $\sigma_1, \sigma_2 \in \Sigma_i$, we say that σ_1 **strictly dominates** σ_2 if

$$\forall \vec{\sigma_{-i}} \in \Sigma_{-i} \qquad u_i(\vec{\sigma_{-i}}, \sigma_1) > u_i(\vec{\sigma_{-i}}, \sigma_2)$$

Thus, σ_1 is **always strictly a better choice** than σ_2, no matter what the others choose. . . so:

a rational agent will never play a strictly dominated strategy.

Iterated Elimination of Strictly Dominated Strategies

We can thus “simplify” games by **deleting** dominated strategies.

Suppose we are given a game G. Then:

1. Let $G_0 = G$ and let $t = 0$
2. Does G_t contain any strictly dominated strategies? If not, we are done: return G_t.
3. Delete any strictly dominated strategies from G_t to obtain a new (simpler) game G_{t+1}, set $t = t + 1$, and go to step (2).

The Outcome of IEDS

- Suppose that after IEDS, we have a **single** outcome remaining...
- ...then we say the game is **dominance-solvable**.

Lemma

If a game G is dominance-solvable, then the unique outcome of the game according to IEDS is the unique pure-strategy Nash equilibrium of G.

Proof: exercise.

Back to the Running Example

1 \ 2	L	R
T	1 1	0 1
B	0 0	1 0

- B is dominated by T, so delete B.
- R is then dominated by L, so delete R.
- Only outcome remaining is (T,L).
- The game is thus dominance solvable.

An Example

1 \ 2	L	C	R
T	4, 3	5, 1	6, 2
M	2, 1	8, 4	3, 6
B	3, 0	9, 6	2, 8

An Example

1 \ 2	L	C	R
T	4, 3	5, 1	6, 2
M	2, 1	8, 4	3, 6
B	3, 0	9, 6	2, 8

Observe that C is dominated by R, so delete it

An Example

1 \ 2	L	R
T	4, 3	6, 2
M	2, 1	3, 6
B	3, 0	2, 8

An Example

1 \ 2	L	R
T	4, 3	6, 2
M	2, 1	3, 6
B	3, 0	2, 8

Here, both M and B are dominated by T

An Example

1 \ 2	L	R
T	4, 3	6, 2

In this game, R is dominated by L, and with one final elimination, the outcome of the game is (T, L).

Evaluating IEDS

- IEDS guarantees a solution exists
- Does not guarantee solution is unique
- Often fails to make any useful predictions (all outcomes survive)
- Powerful when it can be applied
- Hinges on **common knowledge of rationality**

Part XVII

Computing Pure NE

An Algorithm for Computing Pure NE

- In each **column**, draw a box around the utilities of the row player corresponding to the best choice for that player (i.e., the largest blue number(s) in each column)
- In each **row**, draw a box around the utilities of the column player corresponding to the best choice for that player (i.e., the largest red number(s) in each row)
- **Any cell with both payoffs boxed is a NE**
- **Any row with all one player's payoffs boxed is a DS; similarly for columns**

Nash Equilibrium Analysis

Consider the defect column:

1 \ 2	D	C
D	$\boxed{-2}$, -2	0, -3
C	-3, 0	-1, -1

Nash Equilibrium Analysis

Consider the cooperate column:

1 \ 2	D	C
D	$\boxed{-2}$, -2	$\boxed{0}$, -3
C	-3, 0	-1, -1

Nash Equilibrium Analysis

Consider the defect row:

1 \ 2	D	C
D	$\boxed{-2}$ $\boxed{-2}$	-3 $\boxed{0}$
C	0 -3	-1 -1

Nash Equilibrium Analysis

Consider the cooperate row:

1 \ 2	D	C
D	$\boxed{-2}$ $\boxed{-2}$	-3 $\boxed{0}$
C	$\boxed{0}$ -3	-1 -1

Computing Pure Strategy Nash Equilibrium

- Simplest approach to computing pure NE is **exhaustive search**.
- Assume we can compute the value $u_i(\vec{\sigma})$ in unit time.
- The exhaustive search will take time $O(|\Sigma_1 \times \cdots \times \Sigma_n|)$, i.e., exponential in number of agents.

Exhaustive Search for Pure Nash Equilibrium

(For finite games)

```
for σ⃗ ∈ Σ_1 × ··· × Σ_n do
    found = ⊤
    for i ∈ N do
        for σ'_i ∈ Σ_i do
            if u_i(σ⃗_{-i}, σ'_i) > u_i(σ⃗) then
                found ← ⊥
            end if
        end for
    end for
    if found then
        return σ⃗
    end if
end for
return "no pure NE found"
```

Myopic Best Response for Pure Nash Equilibrium

- An alternative can be more efficient in some cases.
- In **myopic best response** we search for a solution with unhappy players flipping their strategies to a best responses.
- In some cases, MBR works well; but not guaranteed.
- If it terminates, it gives a pure NE; but it is not guaranteed to terminate.

Myopic Best Response for Pure Nash Equilibrium

$\vec{\sigma} \leftarrow$ random element of $\Sigma_1 \times \cdots \times \Sigma_n$
while exists player i who is not playing best response in $\vec{\sigma}$ **do**
    $\sigma'_i \leftarrow$ an element of $BR_i(\vec{\sigma})$
    $\vec{\sigma} \leftarrow (\vec{\sigma}_{-i}, \sigma'_i)$
end while
return $\vec{\sigma}$

An Example where Myopic Best Response Fails

1 \ 2	L	C	R
T	-1, 1	1, -1	-2, -2
M	1, -1	-1, 1	-2, -2
B	-2, -2	-2, -2	2, 2

What happens if we start myopic best response at (T, L)?

- A natural question:

  are there classes of games in which
  pure NE are **guaranteed** to exist?

- One natural class of games satisfying this property is **potential games**.
- **Congestion games** are an important class of potential games that have real-world significance.

Potential Games

More accurately: *exact* potential games

A game

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

is said to be a **potential game** if there exists a function

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

such that

1. for all players $i \in N$,
2. for all strategy profiles $\vec{\sigma}$,
3. for all strategies $\sigma_i \in \Sigma_i$ and $\sigma_i' \in \Sigma_i$

we have

$$u_i(\vec{\sigma}_{-i}, \sigma_i) - u_i(\vec{\sigma}_{-i}, \sigma_i') \quad = \quad P(\vec{\sigma}_{-i}, \sigma_i) - P(\vec{\sigma}_{-i}, \sigma_i')$$

Potential Games

Theorem

Every (finite) potential game has a pure strategy Nash equilibrium.

Proof.

1. Find a $\vec{\sigma}^*$ that maximises the value of P.
   (Since the games are finite, such a $\vec{\sigma}^*$ is guaranteed to be exist, though need not be unique.)
2. Claim: $\vec{\sigma}^*$ is a pure NE.

$\square$

Part XVIII

Computational Considerations

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[6]

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

A Boolean game is a structure

$$G = (N, \Phi, \Phi_1, \ldots, \Phi_n, \gamma_1, \ldots, \gamma_n)$$

where:

- $N = \{1, \ldots, n\}$ is the set of players;
- $\Phi = \{p, q, \ldots\}$ is a finite non-empty set of **Boolean variables**;
- Φ_i is the **set of variables under the control of player** *i*; and
- γ_i is a propositional logic formula over Φ – the **goal** of agent *i* (the **specification** for *i*)

Boolean games provide a compact representation of **strategies** and **utility functions**.

Strategies and Utilities

- A strategy for agent i is an assignment

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

  Agent i chooses a value for all its variables.
- A strategy profile defines a complete valuation $\vec{\sigma} : \Phi \to \mathbb{B}$. We write $\vec{\sigma} \models \varphi$ to mean that φ is satisfied by the valuation corresponding to $\vec{\sigma}$
- The utility of strategy profile $(\vec{\sigma})$ to player i is:

$$u_i(\vec{\sigma}) = \begin{cases} 1 & \text{if } \vec{\sigma} \models \gamma_i \\ 0 & \text{otherwise.} \end{cases}$$

- We can then define NE in the standard way.

An Example

$$
\begin{aligned}
N &= \{1, 2\} \\
\Phi_1 &= \{p\} \\
\Phi_2 &= \{q, r\} \\
\gamma_1 &= q \\
\gamma_2 &= q \vee r
\end{aligned}
$$

- How many pure strategies does each player have?
- How many strategy profiles are there?
- How many outcomes are NE?
- Who gets their goal achieved in equilibrium?

Another Example

Matching pennies as a Boolean game

Suppose:

$$\begin{array}{rcl} N & = & \{1,2\} \\ \Phi_1 & = & \{p\} \\ \Phi_2 & = & \{q\} \\ \gamma_1 & = & p \leftrightarrow q \\ \gamma_2 & = & \neg(p \leftrightarrow q) \end{array}$$

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.

NE Membership is co-NP-complete

Work with the complement problem, of verifying that some player has a beneficial deviation.

- **Membership of NP**: Guess a player i and strategy σ_i' and verify that i does better with σ_i' than their component of $\vec{\sigma}$.
- **NP Hardness**: Reduce SAT. Given SAT instance φ define 1-player game with $\gamma_1 = \varphi \wedge z$ where z is a new variable. Define strategy σ_1 which sets all variables to false. φ is then satisfiable iff i has a beneficial deviation from σ_1.

Non-Emptiness is Σ_2^p-complete

Membership

The game has an NE iff the following **Quantified Boolean Formula** is true:

$$\exists \Phi \bigwedge_{i \in N} ((\exists \Phi_i \, \gamma_i) \rightarrow \gamma_i) \tag{7}$$

The formula is a Quantified Boolean Formula with two quantifiers ($\exists\forall$), i.e., an instance of $\text{QBF}_{2,\exists}$, whose truth can be checked in Σ_2^p.

Non-Emptiness is Σ_2^p-complete

Hardness

Reduce $\text{QBF}_{2,\exists}$ to the problem of non-emptiness in a 2-player Boolean games.
Suppose $\exists X \forall Y \psi(X, Y)$ is the $\text{QBF}_{2,\exists}$ instance.
Define a game with:

- $\Phi_1 = X \cup \{x\}$ and $\gamma_1 = \psi(X, Y) \vee (x \leftrightarrow y)$
- $\Phi_2 = Y \cup \{y\}$ and $\gamma_2 = \neg\psi(X, Y) \wedge \neg(x \leftrightarrow y)$

Claim: the game has a NE iff $\exists X \forall Y \psi(X, Y)$ is true.

Non-Emptiness is Σ_2^p-complete

Assume $\exists X \forall Y \psi(X, Y)$ is true

- Then 1 can assign values to variables X such that $\psi(X, Y)$ is true no matter what values are assigned to Y.
- This assignment guarantees player 1's goal γ_1 is satisfied.
- Player 2 has no benefical deviation.

$\therefore$ the game has an NE (which satisfies γ_1).

Non-Emptiness is Σ_2^p-complete

Assume $\exists X \forall Y \psi(X, Y)$ is false

- Then the following formula is true:

$$\forall X \exists Y \neg\psi(X, Y)$$

- Consider any assignment of values to X, Y. It will either satisfy γ_1 or γ_2 since they are the negation of each other.
- If the outcome satisfies γ_1 then player 2 will have a beneficial deviation by the truth of the formula above.
- If the outcome satisfies γ_2 then player 1 will have a beneficial deviation to make $x \vee y$ true.

$\therefore$ the game has no NE.

Computational Game Theory

Lecture 4: Mixed Strategies and Nash's Theorem

UNIVERSITY OF OXFORD

DEPARTMENT OF COMPUTER SCIENCE

Part XIX

Mixed Strategies

Pure Nash Equilibria Don't Always Exist

Recall the game of **matching pennies**:

1 \ 2	Heads	Tails
Heads	1, −1	−1, 1
Tails	−1, 1	1, −1

No pair of strategies forms a pure NE in matching pennies.

Nash Equilibria in Mixed Strategies

1 \ 2	Heads	Tails
Heads	1, -1	-1, 1
Tails	-1, 1	1, -1

- The solution is to allow **mixed strategies**:
  play “heads” and “tails” with equal probability
- To verify it is NE:
  - if player 1 puts more weight on H than T
    ... then best response by 2 is to play T with certainty
  - if player 1 puts more weight on T than H
    ... then best response by 2 is to play H with certainty

Mixed Strategies

A mixed strategy μ for i is a probability distribution over the **pure strategies** Σ_i, hence has the form

play σ_1 with probability p_1
play σ_2 with probability p_2
$\cdots$
play σ_k with probability p_k.

which must satisfy probability constraints:

$$\begin{array}{rcll} p_1 + p_2 + \cdots + p_k & = & 1 & \\ p_i & \in & [0,1] & \text{for all } 1 \leq i \leq k \end{array}$$

Let $MS_i = \Delta\Sigma_i$ be the set of all mixed strategies for player i.

☞ **We are in the domain of expected utility.** ☜

Nash's Theorem

A game is finite if Σ_i is finite for all $i \in N$.

Theorem (Nash, 1950)

Every finite game has a Nash equilibrium in mixed strategies.

- Guarantees the existence of NE
- But what about **computing** NE…?

Part XX

The Indifference Principle Part 1: 2×2 Games

Two Player Games

- Mixed Nash equilibria for 2×2 games are easy to compute
- We need some more definitions and results to get there...
- The technique we use is called the **indifference principle**

The Supports of a Mixed Strategy

The **support** of a mixed strategy

$$\mu_i : \Sigma_i \to [0, 1]$$

is the set of pure strategies played with +ve probability in μ_i:

$$\mathit{supp}(\mu_i) = \{\sigma \mid \mu_i(\sigma) > 0\}$$

A mixed strategy μ_i is **fully mixed** if

$$\mathit{supp}(\mu_i) = \Sigma_i$$

i.e., all pure strategies are played with +ve probability.

A Generic 2 × 2 Games

We will work with the a generic 2×2 game:

1 \ 2	L	R
T	v_1^1, v_1^2	v_2^1, v_2^2
B	v_3^1, v_3^2	v_4^1, v_4^2

- The **superscript** identifies the player (1 or 2)
- The **subscript** identifies the cell (1 to 4)

Mixed Strategies in 2*x*2 Games

1 \ 2	L	R
T	v_1^1, v_1^2	v_2^1, v_2^2
B	v_3^1, v_3^2	v_4^1, v_4^2

Represent a mixed strategy for player 1 as a value $p \in [0, 1]$:

- play T with probability p
- play B with probability $1 - p$

Represent a mixed strategy for player 2 as a value $q \in [0, 1]$:

- play L with probability q
- play R with probability $1 - q$

A mixed strategy profile is a pair (p, q)

Expected Utility in Mixed Strategies

1 \ 2	L	R
T	v_1^1, v_1^2	v_2^1, v_2^2
B	v_3^1, v_3^2	v_4^1, v_4^2

Suppose $(p, q) \in (0, 1)^2$ is a pair of mixed strategies. Define:

$$\begin{aligned}
EU_1(T, q) &= (v_1^1 \times q) + (v_2^1 \times (1 - q)) \\
EU_1(B, q) &= (v_3^1 \times q) + (v_4^1 \times (1 - q)) \\
EU_2(L, p) &= (v_1^2 \times p) + (v_3^2 \times (1 - p)) \\
EU_2(R, p) &= (v_2^2 \times p) + (v_4^2 \times (1 - p))
\end{aligned}$$

Indifference Principle for 2x2 Games

Theorem

If $(p, q) \in (0, 1)^2$ *is a mixed strategy Nash equilibrium in the generic* 2×2 *game then:*

$$\begin{aligned} EU_1(T, q) &= EU_1(B, q) \quad \text{and} \\ EU_2(L, p) &= EU_2(R, p). \end{aligned}$$

This is a **special case** of a **general result**, called the **indifference principle**: The expected payoff you would get from all pure strategies in the support of a NE is the same.

Algorithm for Computing Mixed NE in $2x2$ Games

1. Check for **pure** NE. A rule of thumb is that games contain an **odd** number of NE: if you find an even number of pure NE, look for mixed NE...
2. Consider the following equality:

   $$EU_1(T, q) = EU_1(B, q)$$

   Find solutions for q.
3. Then consider the following equality:

   $$EU_2(L, p) = EU_2(R, p)$$

   Find solutions for p
4. Any pair of solutions (p, q) defines a mixed NE

Since we are dealing with **linear equalities**, we can solve them in polynomial time.

Computing Mixed NE in Matching Pennies

1 \ 2	Heads	Tails
Heads	$1, -1$	$-1, 1$
Tails	$-1, 1$	$1, -1$

First we find q:

$$\begin{aligned}
EU_1(Heads, q) &= EU_1(Tails, q) \\
(v_1^1 \times q) + (v_2^1 \times (1-q)) &= (v_3^1 \times q) + (v_4^1 \times (1-q)) \\
(1 \times q) + (-1 \times (1-q)) &= (-1 \times q) + (1 \times (1-q)) \\
2q - 1 &= 1 - 2q \\
4q &= 2 \\
q &= 0.5
\end{aligned}$$

Computing Mixed NE in Matching Pennies

Now we find p:

$$\begin{aligned}
EU_2(Heads, p) &= EU_2(Tails, p) \\
(v_1^2 \times p) + (v_3^2 \times (1-p)) &= (v_2^2 \times p) \;\; + \;\; (v_4^2 \times (1-p)) \\
(-1 \times p) + (1 \times (1-p)) &= (1 \times p) + (-1 \times (1-p)) \\
1 - 2p &= 2p - 1 \\
2 &= 4p \\
p &= 0.5
\end{aligned}$$

So $(0.5, 0.5)$ is a mixed NE in matching pennies game.

Part XXI

The Indifference Principle Part 2: Finite Two Player Games

The Indifference Principle

The 2×2 instance is a special case of a general result.

Theorem (Indifference Principle)

If mixed strategy profile $\vec{\mu} = (\mu_1, \ldots, \mu_n)$ is a NE then:

1. *for all $i \in N$, and*
2. *for all $\sigma_1, \sigma_2 \in supp(\mu_i)$, we have*

$$EU_i(\sigma_1, \vec{\mu}_{-i}) = EU_i(\sigma_2, \vec{\mu}_{-i})$$

This suggests an approach for computing mixed NE in 2 player games with > 2 strategies; the difficulty is knowing the support.

The Support Enumeration Method (SEM)

NB: For two player games

- Now, **if we know the support of a NE**, then we can find that NE by solving a **linear program** (technically, a **linear feasibility program**)
- Given game G with supports defined by $S = (S_1, S_2)$ (so $S_i \subseteq \Sigma_i$ is the support of the NE for player i), let the LFP $SEP(G, S_1, S_2)$ be as follows…

The Linear Feasibility Program $SEP(G, S_1, S_2)$

$$\begin{aligned}
\sum_{\sigma_2 \in S_2} p_2(\sigma_2) u_1(\sigma_1, \sigma_2) &= v_1 \quad \forall \sigma_1 \in S_1 && (8)\\
\sum_{\sigma_2 \in S_2} p_2(\sigma_2) u_1(\sigma_1, \sigma_2) &\leq v_1 \quad \forall \sigma_1 \in (\Sigma_1 \setminus S_1) && (9)\\
\sum_{\sigma_1 \in S_1} p_1(\sigma_1) u_2(\sigma_1, \sigma_2) &= v_2 \quad \forall \sigma_2 \in S_2 && (10)\\
\sum_{\sigma_1 \in S_1} p_1(\sigma_2) u_2(\sigma_1, \sigma_2) &\leq v_2 \quad \forall \sigma_2 \in (\Sigma_2 \setminus S_2) && (11)\\
\sum_{\sigma_i \in S_i} p_i(\sigma_i) &= 1 \quad \forall i \in \{1, 2\} && (12)\\
p_i(\sigma_i) &> 0 \quad \forall i \in \{1, 2\}, \sigma_i \in S_i && (13)\\
p_i(\sigma_i) &= 0 \quad \forall i \in \{1, 2\}, \sigma_i \in (\Sigma_i \setminus S_i) && (14)
\end{aligned}$$

Unknowns are variables $p_i(\sigma_i)$ and v_i. Values $u_i(\cdots)$ are constants.

A Exponential Time Algorithm Using Support Enumeration

```
for i = 2 to |Σ_1| do
    for j = 2 to |Σ_2| do
        for each S_1 ⊆ Σ_1 such that |S_1| = i do
            for each S_2 ⊆ Σ_2 such that |S_2| = j do
                if SEP(G, S_1, S_2) has a solution then
                    return S_1, S_2, p_1, p_2, v_1, v_2
                end if
            end for
        end for
    end for
end for
```

Part XXII

Nash's Theorem in the 2×2 Case

Best Response Functions

Best response functions with mixed strategies are generalisations of pure strategy case:

$$BR_i : MS_1 \times \cdots \times MS_n \rightarrow \mathbf{2}^{MS_i}$$

where

$$BR_i(\vec{\mu}) = \arg \max_{\mu_i \in MS_i} EU_i(\vec{\mu}_{-i}, \mu_i)$$

The best response function of the game is:

$$BR(\vec{\mu}) = BR_1(\vec{\mu}) \times \cdots \times BR_n(\vec{\mu})$$

NE are Fixed Points of the Best Response Function

- Recall that...
  - $s \in S$ is a fixed point of $f : S \to S$ if $f(s) = s$.
  - $s \in S$ is a fixed point of $f : S \to \mathbf{2}^S$ if $s \in f(s)$.
- Then by definition of NE we have:

Lemma

$$\vec{\mu} \in NE(G)$$

iff

$$\vec{\mu} \in BR(\vec{\mu})$$

A Graphical "Proof" for the 2×2 Case

- We can "prove" Nash's theorem in the 2×2 case by plotting the best response functions BR_1 and BR_2 against each other.
- To illustrate, we work with the following game:

1 \ 2	L	R
T	1, 5	0, 0
B	0, 0	2, 1

- Solving, we find a unique mixed NE with

$$(p, q) = \left(\frac{1}{6}, \frac{2}{3}\right)$$

A Graphical "Proof" for the 2×2 Case

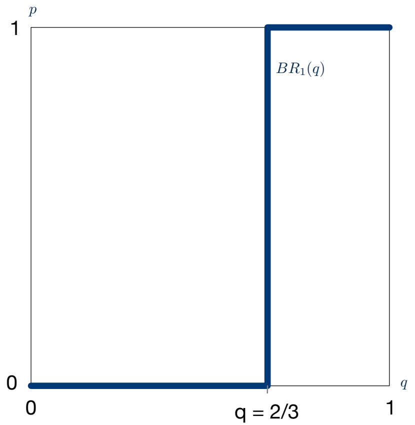
1 \ 2	L	R
T	1, 5	0, 0
B	0, 0	2, 1

$$BR_1(q) = \begin{cases} \{0\} & \text{if } q < 2/3 \\ [0,1] & \text{if } q = 2/3 \\ \{1\} & \text{if } q > 2/3 \end{cases}$$

$$BR_2(p) = \begin{cases} \{0\} & \text{if } p < 1/6 \\ [0,1] & \text{if } p = 1/6 \\ \{1\} & \text{if } p > 1/6 \end{cases}$$

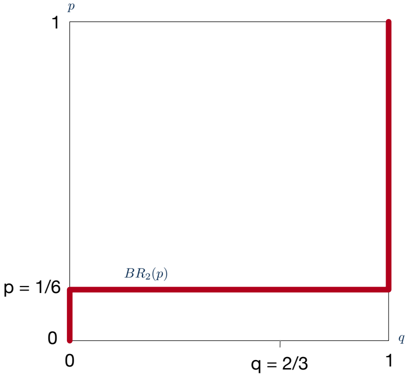
Plotting the Best Response Function $BR_1(q)$

$$BR_1(q) = \begin{cases} \{0\} & \text{if } q < 2/3 \\ [0,1] & \text{if } q = 2/3 \\ \{1\} & \text{if } q > 2/3 \end{cases}$$

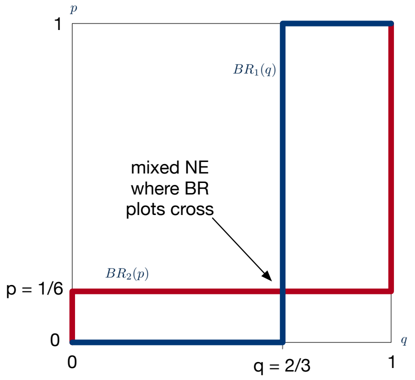


Plotting the Best Response Function $BR_2(p)$

$$BR_2(p) = \begin{cases} \{0\} & \text{if } p < 1/6 \\ [0,1] & \text{if } p = 1/6 \\ \{1\} & \text{if } p > 1/6 \end{cases}$$



Plotting $BR_1(q)$ against $BR_2(p)$



Observation

Any *BR plots in the* 2×2 *case must cross somewhere in the unit square* $\Rightarrow$ *NE in mixed strategies must exist in* 2×2 *case.*

Part XXIII

Nash's Theorem

Fixed Point Theorems

- The key to Nash's result are a class of results in algebraic topology, known as **fixed point theorems**.
- Fixed point theorems characterise the existence of fixed point in functions with respect to their properties.
- Nash's theorem can be proved via
  - **Brouwer's fixed point theorem** or
  - **Kakutani's fixed point theorem**

  Proof via Kakutani is most direct.

Brouwer's Fixed Point Theorem

Theorem (Brouwer, 1909)

Let S be a convex, bounded, closed set and let $f : S \to S$ be a continuous function from S to itself. Then f has a fixed point.

- **convex**: S does not contain "holes"
- **bounded**: every element is within a "fixed distance" of every other element
- **closed**: contains its own end points
- **continuous**: you can plot the function "without lifting pen from paper".

We will now show these conditions are **necessary** for the existence of a fixed point; the proof that they are **sufficient** is more involved.

Brouwer's Theorem: Convexity

- A set $S \subseteq \mathbb{R}^k$ is convex if it "contains no holes".
- S is convex if for any two elements $A, B \in S$, all points on the straight line connecting A to B are contained in S

Example

Let $S \subseteq \mathbb{R}^2$ define a **circle** and let $f : S \to S$ map every point on the circle to the point 90° anticlockwise. Clearly f is continuous. No fixed point!

Brouwer's Theorem: Boundedness

- A set $S \subseteq \mathbb{R}$ is **bounded** if it has upper and lower bounds, i.e., values x and y such that for all values $z \in \mathbb{R}$, we have $x \leq z$ and $y \geq z$.
- For multiple dimensions, boundedness generalises: all points are within a fixed distance of each other.

Example

Let $S = \mathbb{R}_+$ and define $f : S \to S$ by

$$f(x) = x + 1.$$

No fixed point.

Brouwer's Theorem: Continuity

Intuitively a function f is continuous if it can be plotted without lifting pen from paper.

Example

Let $S = [0, 1]$ and define $f : S \to S$ by

$$f(x) = \begin{cases} 0.7 & \text{if } x \leq 0.5 \\ 0.3 & \text{otherwise.} \end{cases}$$

Discontinutity at $x = 0.5$: no fixed point.

Brouwer's Theorem: Closed Set

- A set is closed if it contains its own end points.
- The set $[0, 1]$ is closed; $(0, 1)$ is not, nor is $(0, 1]$.

Example

Let $S = [0, 1)$, and define

$$f(x) = \frac{x + 1}{2}$$

This function shifts every point to the right, and while $f(x) \to 1$ as $x \to 1$, it does not have a fixed point.

If $S = [0, 1]$, there **is** fixed point: $f(1) = 1$.

A Natural Special Case

Theorem

Every continuous function $f : [0, 1] \to [0, 1]$ *has a fixed point.*

The proof is easy:

1. Plot the function $f(x)$ for x from 0 to 1 inclusive.
2. At some point, the line **must** cross the diagonal line $y = x$. (Now see the need for f to be continuous, and to include end points 0 and 1.)
3. At the point (x, y) where crosses the diagonal, we have $y = f(x) = x$, i.e., a fixed point.

Kakutani's Fixed Point Theorem

Theorem (Kakutani, 1940.)

Suppose S is a non-empty, compact (closed & bounded), and convex subset of $\mathbb{R}^n$, and suppose $f : S \to \mathbf{2}^S$ is such that $f(s)$ is non-empty and convex for all $s \in S$, and that f has a closed graph. Then f has a fixed point.

Nash's Theorem

Theorem (Nash, 1950)

Every finite game G has a Nash equilibrium in mixed strategies.

Proof.

1. The Nash equilibria of G are precisely the fixed points of the game's best response function BR.
2. The game's best response function satisfies the conditions of Kakutani's fixed point theorem.

□

All the work of the proof of Nash's theorem is therefore in showing that the game's best response function satisfies Kakutani's conditions.

From the source...

Any n-tuple of strategies, one for each player, may be regarded as a point in the product space obtained by multiplying the n strategy spaces of the players. One such n-tuple counters another if the strategy of each player in the countering n-tuple yields the highest obtainable expectation for its player against the $n - 1$ strategies of the other players in the countered n-tuple. A self-countering n-tuple is called an equilibrium point.

The correspondence of each n-tuple with its set of countering n-tuples gives a one-to-many mapping of the product space into itself. From the definition of countering we see that the set of countering points of a point is convex. By using the continuity of the pay-off functions we see that the graph of the mapping is closed. The closedness is equivalent to saying: if $P_1, P_2, \ldots$ and $Q_1, Q_2, \ldots, Q_n, \ldots$ are sequences of points in the product space where $Q_n \to Q$, $P_n \to P$ and Q_n counters P_n then Q counters P.

Since the graph is closed and since the image of each point under the mapping is convex, we infer from Kakutani's theorem[1] that the mapping has a fixed point (i.e., point contained in its image). Hence there is an equilibrium point.

Computational Game Theory

Lecture 5: Dynamic Games

UNIVERSITY OF OXFORD

DEPARTMENT OF COMPUTER SCIENCE

Part XXIV

Introduction

Introduction

- Strategic form games assume that **players make just one choice**, and that **this move is made ignorance of the choices of others**
- “Simultaneous moves” is really an **informational** assumption, not a **temporal** one
- In many settings, games have a much richer dynamic and informational structure.
- In this lecture study three classes of dynamic games:
  - **extensive form games**
  - **iterated games**
  - **evolutionary games**

Part XXV

Extensive Form Games

Extensive Form Games

- Many games of interest involve **multiple moves**
- Information may or may not be available about previous moves
- **Extensive form games** model scenarios with this structure
  - **games of perfect information**: players know exactly how the current state of the game was reached
  - **games of imperfect information**: players may be uncertain about previous moves, may not know how the reached the current game state
  - **games of imperfect recall**: players may forget things they knew previously (even their own moves)
- (We restrict ourselves to games with **no chance moves**.)

Reminder: Trees

A **tree**, $T = (V, E \subseteq V \times V, v_0)$, is a directed acyclic graph where:

1. there is a single vertex with no incoming edges
   — the **root**, denoted v_0;
2. there is a path from the root to every other vertex;
3. every non-root vertex has a single incoming edge.

Reminder: Trees

- The **children** of v are denoted by $\mathit{children}(v, (V, E))$:

$$\mathit{children}(v, (V, E)) = \{v' \mid (v, v') \in E\}$$

- The **leaves** of T, denoted $\mathit{leaves}((V, E))$, are nodes with no children:

$$\mathit{leaves}((V, E)) = \{v \mid v \in V \text{ and } \mathit{children}(v) = \emptyset\}$$

- Non-leaf vertices are called **interior vertices**:

$$\mathit{interior}((V, E)) = V \setminus \mathit{leaves}((V, E))$$

Game Trees

- A finite tree structure $T = (V, E, v_0)$, with vertices V, edges $E \subseteq V \times V$, and root v_0
- Leaves are **labelled** with payoffs for each player:

$$u_i : \mathit{leaves}(T) \to \mathbb{R}$$

- Interior nodes of T are **decision nodes**, and are labelled with the player who makes a move at that point:

$$\mathit{owner} : \mathit{interior}((V, E)) \to N$$

- Each **edge** corresponds to a **move** or **action** that can be made by that player.
- The player at the root of the tree moves first.

Game Trees

- Let V_i denote the decision nodes for player i:

$$V_i = \{v \mid owner(v) = i\}$$

- Each edge $(v, v') \in E$ is labelled with an action $a(v, v')$
- Let $A(v) = \{a(v, v') \mid (v, v') \in E\}$ be the actions available at vertex v
- We require that $a(v, v') = a(v, v'')$ implies $v' = v''$
  (What does this condition mean?)

Game Trees

- Let V_i denote the decision nodes for player i:

$$V_i = \{v \mid owner(v) = i\}$$

- Each edge $(v, v') \in E$ is labelled with an action $a(v, v')$
- Let $A(v) = \{a(v, v') \mid (v, v') \in E\}$ be the actions available at vertex v
- We require that $a(v, v') = a(v, v'')$ implies $v' = v''$
(What does this condition mean?)
- Let A_i be the total set of actions available to i in the game:

$$A_i = \bigcup_{v \in V_i} A(v)$$

Extensive Form Games of Perfect Information

$$G = (N, (V, E, v_0), owner, a, u_1, \ldots, u_n)$$

where:

- $N = \{1, \ldots, n\}$ is the set of players;
- (V, E, v_0) is a finite tree with vertex set V, edge set $E \subseteq V \times V$, and root $v_0 \in V$;
- $owner : interior((V, E)) \to N$ specifies the owner of each decision node;
- $a : E \to A$ associates each edge $(v, v') \in E$ with an action;
- $u_i : leaves((V, E)) \to \mathbb{R}$ is i's utility function.

These components must satisfy the constraints stated earlier.

Strategies in Extensive Form Games

A **(pure) strategy**, σ_i, for player $i \in N$ is a function that selects an action (move) for all of i's decision nodes:

$$\sigma_i : V_i \to A_i$$

such that

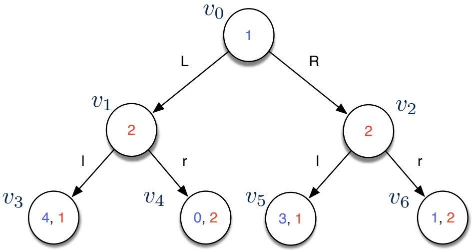
$$\sigma_i(v) \in A(v)$$

Let Σ_i be the set of pure strategies for $i \in N$; define strategy profiles, NE, etc, as in normal form games.

Observation

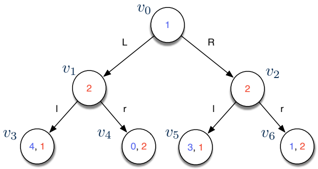
Every strategy profile $\vec{\sigma}$ induces a unique path in the game tree from the root to a leaf node, which is the outcome of the game under that strategy profile. If the strategy profile is an equilibrium, then the path is an ***equilibrium path****.*

Example Game Tree



Two players: $N = \{1, 2\}$.
First player to move is 1; she can perform either *L* or *R* moves.

Pure Strategies for Extensive Form Games

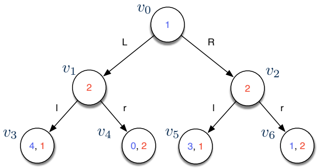


IMPORTANT: A strategy for *i* defines a choice for **all** decision nodes V_i

There are **two** pure strategies for 1:

- $\sigma_1^1(v_0) = L$
- $\sigma_1^2(v_0) = R$

Pure Strategies for Extensive Form Games



There are **four** pure strategies for 2:

- $\sigma_2^1(v_1) = l \quad \sigma_2^1(v_2) = l$
- $\sigma_2^2(v_1) = l \quad \sigma_2^2(v_2) = r$
- $\sigma_2^3(v_1) = r \quad \sigma_2^3(v_2) = l$
- $\sigma_2^4(v_1) = r \quad \sigma_2^4(v_2) = r$

- Use backward induction to label every node with payoff profile that would be achieved in equilibrium (**dynamic programming**).
- Repeat the following:
  - For each decision node $v \in V$:
    - If all the children of v have been labelled with a payoff profile, then label v with a payoff profile from a child that maximises the payoff of the player making the decision at that node. (If there is a choice, choose arbitrarily.)

  until all vertices have been labelled with payoff profiles.

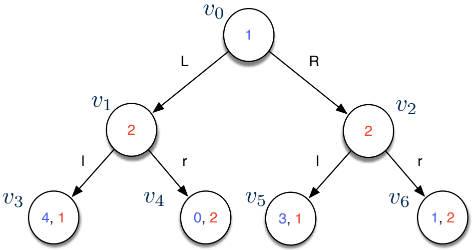
Zermelo's Algorithm

Solving Extensive Form Games with Backwards Induction

$done \leftarrow leaves((V, E))$
while $done \neq V$ **do**
  $next \leftarrow \{v' \in V \setminus done \mid children(v', (V, E)) \subseteq done\}$
  for $v \in next$ **do**
    $i \leftarrow owner(v)$
    $C \leftarrow children(v, (V, E))$
    $O \leftarrow \arg\max_{v' \in C} u_i(v')$ // `optimal choices`
    $v'' \leftarrow$ any element of O
    for $j \in N$ **do**
      $u_j(v) \leftarrow u_j(v'')$ // `back utilities up`
    end for
    $done \leftarrow done \cup \{v\}$ // `we have processed` v
  end for
end while

Illustrating Zermelo's Algorithm

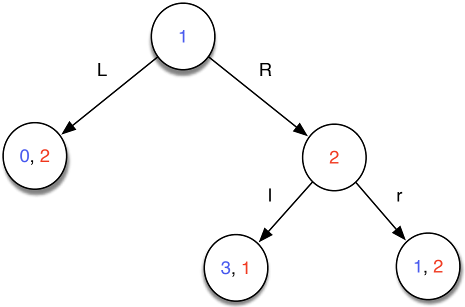
Recall our game:



To illustrate the algorithm, we **delete** parts of the game tree that we have already "processed".

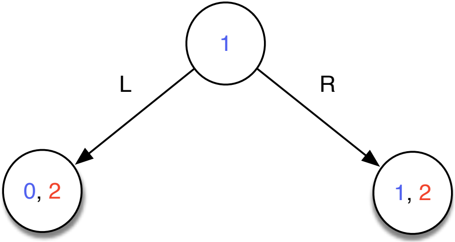
Initially, start with 2's bottom left choice: given a choice between 1 and 2, she will choose 2, i.e., move "r".

Illustrating Zermelo's Algorithm



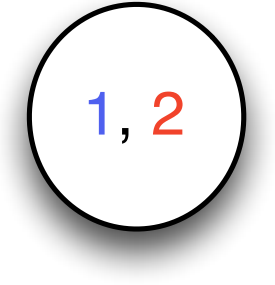
Now consider 2's bottom right choice: given a choice between 1 and 2, she will choose 2, i.e., move "r".

Illustrating Zermelo's Algorithm



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

Illustrating Zermelo's Algorithm



So, player 1 receives 1 in equilibrium, while player 2 receives 2. We write an equilibrium in an extensive form game by listing the actions for each player in turn. In this case: (R, r).

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

Properties of Extensive Form Games

Theorem

1. *Every extensive form game (with perfect information and no chance moves) has a NE in pure strategies.*
2. *Pure strategy NE in extensive form games can be computed in polynomial time with Zermelo's algorithm.*
3. *If no two leaf nodes have the same utility for any player, then the NE is unique.*

Proof: Zermelo's algorithm.

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[7]
  - Computing optimal policies in Markov decision processes via "value iteration"[8]

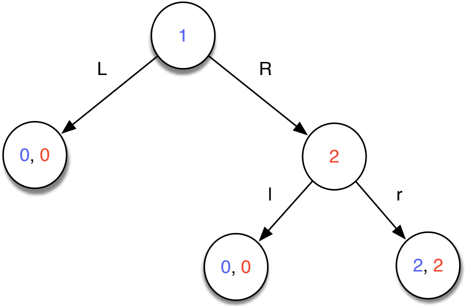
[7] E. M. Clarke, O. Grumberg, and D. Peled. *Model Checking*, MIT Press, 1999. pages 35–39.

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

Part XXVI

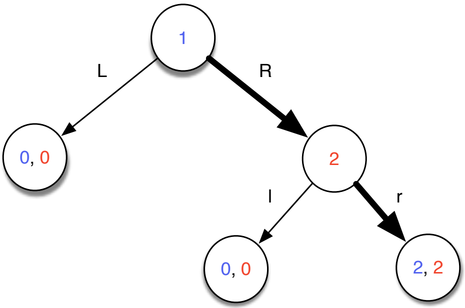
Subgame Perfect Nash Equilibrium

An Extensive Form Game with a Paradoxical NE



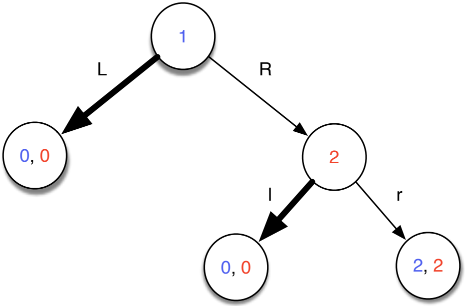
What does Zermelo give when applied to this?

An Extensive Form Game with a Paradoxical NE



Zermelo tells us that (R, r) is a NE, which makes sense.

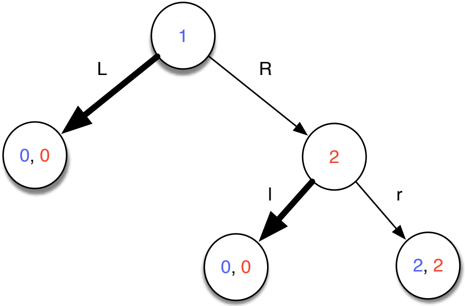
An Extensive Form Game with a Paradoxical NE



But (L, l) is **also** a NE:

- if 1 plays L then 2 will get 0 whatever she does.
  $\Rightarrow$ l is a best response to L
- if 2 chooses l then 1 has a choice of choosing L and receiving 0, or playing R and receiving 0.
  $\Rightarrow$ L is a best response to l

An Extensive Form Game with a Paradoxical NE

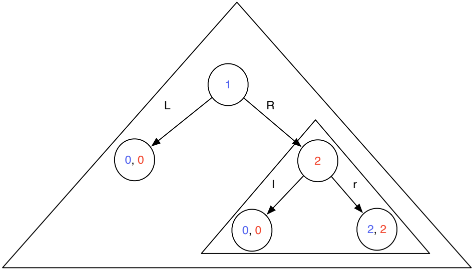


- Here, 2 is **threatening** to play *l*
- . . . but is this threat **credible**? If 2 is ever called on to make a choice, we would be irrational to choose *l*!
- This is a weakness of NE in extensive form games.
- We need a refinement of NE, due to Reinhard Selten, called **subgame perfect Nash equilibrium (SPNE)**

Subgames

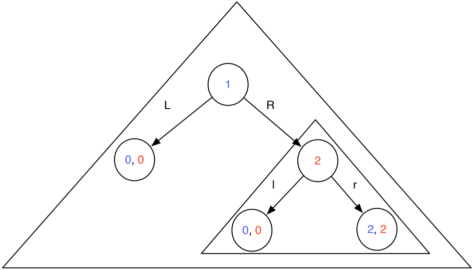
- To define SPNE, we need the notion of a **subgame**
- The **subgames of an extensive form game** G are the games induced by each decision node of G (with strategies, etc, restricted appropriately)
- (Remember that G is a subgame of itself.)

Subgames Illustrated



In this example, the game has just two sub-games.

Subgame Perfect Nash Equilibrium (SPNE)



- A strategy profile $\vec{\sigma} = (\sigma_1, \ldots, \sigma_n)$ is a **subgame perfect Nash equilibrium** of a game G if it is a Nash equilibrium in each subgame G' of G
- Observe that (L, l) is not a SPNE, because l is not a NE of the subgame induced by 2's decision node
- However, (R, r) **is** a SPNE

Computing SPNE

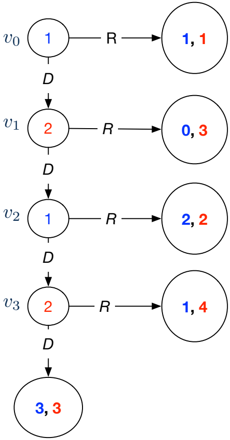
Zermelo’s Algorithm Revisited

Theorem

1. *Every extensive form game (of perfect information and no chance nodes) has a SPNE*
2. *SPNE for extensive form games can be computed in polynomial time using Zermelo’s algorithm.*

The Centipede Game

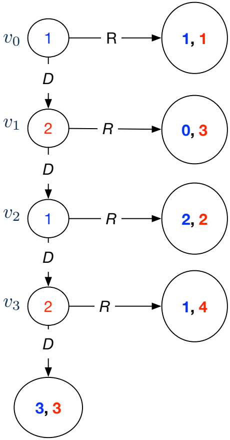
A Game With a Counterintuitive SPNE



What does Zermelo say?

The Centipede Game

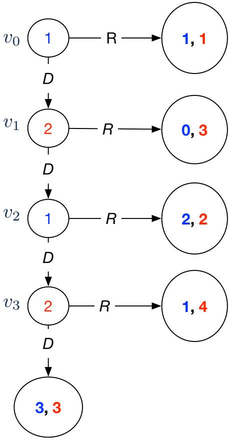
A Game With a Counterintuitive SPNE



What does Zermelo say? SPNE says that first player moves *R* and game ends immediately.

The Centipede Game

A Game With a Counterintuitive SPNE



- In practice, people manage to move *D* for a few rounds before someone moves *R*, leaving them both better off
- A SPNE with poor social welfare

From Extensive Form to Strategic Form Games

- We can represent extensive form games as strategic form games, and solve them using techniques that we use for these.
- Recall that a strategy profile $\vec{\sigma}$ in an extensive form game uniquely determines a leaf node, with payoffs for each player.
- So, for each player i, define u_i
- Let $\Sigma_i = \{\sigma_i^1, \ldots, \sigma_i^k\}$ be the pure strategies for player $i \in N$
- This defines a strategic form game
- Note that we get an **exponential blowup**: Σ_i may be exponentially larger than the original game tree

Part XXVII

Imperfect Information in Extensive Form Games

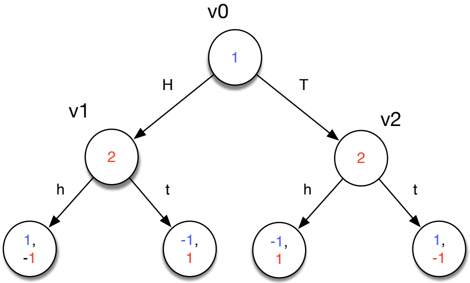
What do player’s know?

- In the extensive form game model we have looked at so far, all players have **perfect information** about the game. In particular, they know all the moves that have been made to date.
- This is often unrealistic!
- A variation of extensive form games allows us to capture **imperfect information**

The Need for Imperfect Information

A Failed Attempt to Represent Matching Pennies

Suppose we try to capture matching pennies as an extensive form game...



- This doesn't work, because when it comes to his move, 2 will know whether 1 has shown heads or tails! Consider:

$$\sigma_2(v) = \begin{cases} t & \text{if } v = v_1 \\ h & \text{if } v = v_2 \end{cases}$$

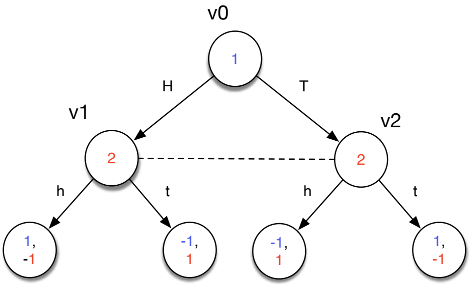
Modelling Imperfect Information

- Partition each players decision nodes into **information sets**
- Let $\mathcal{I}_i$ denote player *i*'s information sets
- If $v \in V_i$ then denote by $[v]$ the information set containing v (note $v \in [v]$)
- Intuition:
  - if $[v] = [v']$ then the decision player does not know whether she is in v or v'
  - she **cannot distinguish** these nodes
- We require that if $[v] = [v']$ then $A(v) = A(v')$
- A strategy in an imperfect information game is then a function that assigns an action to each information set

$$\sigma_i : \mathcal{I}_i \to A_i$$

(We are glossing over some technicalities here. . .)

Matching Pennies as an Imperfect Information Game



- Information sets indicated with a dotted line (but don't draw singletons)

$$\mathcal{I}_2 = \{\{v_1, v_2\}\} \qquad \mathcal{I}_1 = \{\{v_0\}\}$$

- Thus, when 2 makes her move, she doesn't know whether 1 chose H or T

Part XXVIII

Randomized Strategies in Extensive Form Games

Mixed and Behavioural Strategies

- Recall that in strategic games, a mixed (randomized) strategy ms_i for player i is a probability distribution over player i's pure strategies Σ_i
- In extensive form games, we have two ways in which we can ramdomize:
  - mixed strategies
  - behavioural strategies

Mixed Strategies

- As in strategic form games, a mixed strategy in an extensive form game is a probability distribution over pure strategies i.e., a probability distribution
- So, denote by $MS_i = \Delta\Sigma_i$ the set of mixed strategies for i

Behavioural Strategies

- An alternative formulation of randomized strategies has players randomizing at each decision node
- For extensive form games of perfect information, a behavioural strategy β_i is then a function

$$\beta_i : V_i \to \Delta A_i$$

such that

$$supp(\beta_i(v)) \subseteq A(v)$$

- These are called **behavioural strategies**

- An obvious question:

How are mixed and behavioural strategies related? Is one kind more "expressive" than the other?

- The answer is that, under certain conditions, they are equivalent.

Kuhn's Theorem

Recall a player i has **perfect recall** if she knows all her previous decisions.

Theorem

In extensive form games with perfect recall

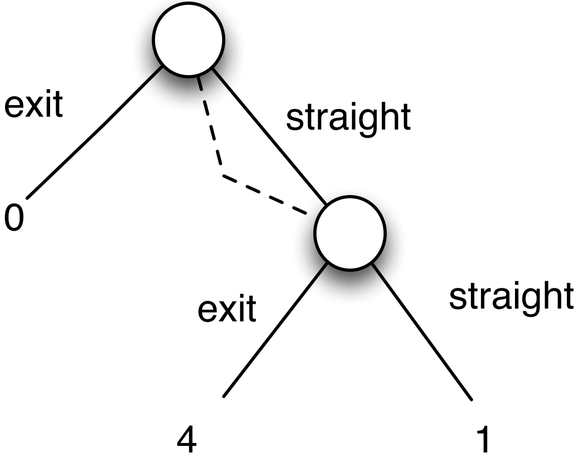
1. *for every mixed strategy there exists a behavioural strategy that yields the same probability distribution over outcomes*
2. *for every behavioural strategy there exists a mixed strategy that yields the same probability distribution over outcomes*

Since perfect information implies perfect recall, the result holds for games of perfect information.

The Forgetful Driver

- An absent minded professor is driving home. It is foggy and hard to see much. The road has two exits, A, and B, which appear after each other.
- Exit A involves a long drive through poor country roads, yielding the driver a utility of 0.
- Exit B is the best: it goes home directly on good roads, yielding a utility of 4.
- If the driver does not exit at B, then she has to drive a fair distance to get home, but not so far as if she exited at A, yielding a utility of 1.
- However, the professor is absent minded, and when she reaches an exit, in the fog she cannot tell whether it is exit A or exit B.

The Forgetful Driver



The Forgetful Driver

- Any pure strategy will yield payoff 0 or 1 (why?)
- Since mixed strategies randomize over pure strategies, any mixed strategy will either exit immediately or drive straight to the end.
- **The only chance to get payoff 4 is to randomize at decision nodes.**

Part XXIX

Iterated Games

Recall the Prisoner’s Dilemma…

An Equilibrium with Undesirable Social Properties

1 \ 2	D	C
D	-2, -2	0, -3
C	-3, 0	-1, -1

Mutual defection (2 years in jail each) is the unique **dominant strategy equilibrium** … although this outcome fails every test of what is a “socially” reasonable outcome.

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. . . ?

Finitely Repeated Prisoner's Dilemma

Analysis via Backwards Induction

- Suppose you know you will play the PD game for n rounds
- Imagine yourself playing the **final round**
- Round n is a **one-shot prisoner's dilemma**: you will defect
- But now consider round $n - 1 \ldots$

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

- The **limit of means** approach involves (intuitively) computing the average payoff over the infinite run
- The value of the infinite run

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

to player i is then

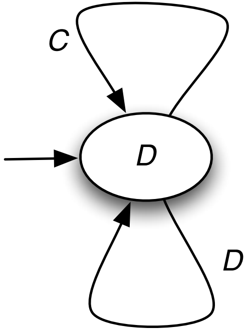
$$\lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T} u_i(\omega_t)$$

- **Problem**: For infinite sequences in general, this value doesn't always converge.

☞ But it **does** if players use **automata strategies**! ☜

Strategies as Finite State Machines

- We represent strategies as **finite state machines with output** – technically, **Moore machines** ("transducers")
- Here is an automaton strategy called "ALLD", which always defects:

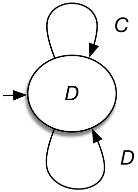


- Value inside a state is the action selected; outgoing arrows are actions of counterpart.

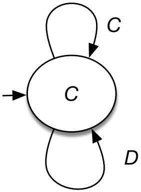
FSM Strategies for the Iterated Prisoner's Dilemma

What do these strategies do?

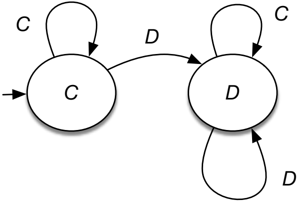
ALLD



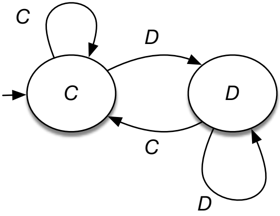
ALLC



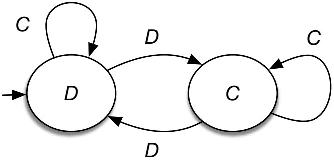
GRIM



TIT-FOR-TAT



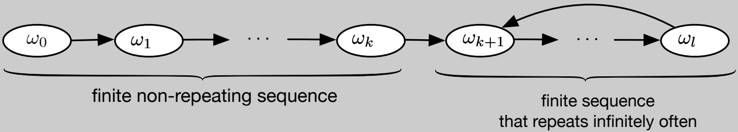
TAT-FOR-TIT



Automata strategies playing against each other

Theorem

Finite machine strategies playing against each other will generate a run of the following form:



Note that non-repeating sequence may be empty; the repeating sequence may be of length one.

The limit of means utility of such a sequence is then simply the average utility over just the ***finite repeating sequence*** $\omega_{k+1}, \ldots, \omega_l$.

So, to find the utility of a player, simply identify the repeating sequence and find the average utility for that player over the finite repeating sequence.

ALLC against ALLC

round:	0	1	2	3	4	$\cdots$	
ALLC:	C	C	C	C	C	$\cdots$	utility = -1
ALLC:	C	C	C	C	C	$\cdots$	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	$\cdots$	utility = -3
ALLD:	D	D	D	D	D	$\cdots$	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	$\cdots$	utility $= -2$
ALLD:	D	D	D	D	D	$\cdots$	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	$\cdots$	utility = -2
ALLD:	D	D	D	D	D	$\cdots$	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	$\cdots$	utility = -1
GRIM:	C	C	C	C	C	$\cdots$	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 **security 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 security 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 she gets her reservation value.

A Worked Example

Consider the following stage game:

1 \ 2	L	R
T	-1, -3	1, 0
M	2, -1	0, 1
B	1, 1	-2, 1

Which outcomes can be sustained as NE in the infinitely repeated stage game?
We first need to identify the **security value** for each player.

A Worked Example

1 \ 2	L	R
T	-1, -3	1, 0
M	2, -1	0, 1
B	1, 1	-2, 1

Define $\bar{u}_i(\sigma_j)$ as follows:

$$\bar{u}_i(\sigma_j) = \max\{u_i(\sigma_i, \sigma_j) \mid \sigma_i \in \Sigma_i\}$$

Thus $\bar{u}_i(\sigma_j)$ is the **largest** utility that *i* could get if *j* plays σ_j

A Worked Example

1 \ 2	L	R
T	-1, -3	1, 0
M	2, -1	0, 1
B	1, 1	-2, 1

- $\bar{u}_1(L) = \max\{-1, 2, 1\} = 2$
- $\bar{u}_1(R) = \max\{1, 0, -2\} = 1$
- $\bar{u}_2(T) = \max\{-3, 0\} = 0$
- $\bar{u}_2(M) = \max\{-1, 1\} = 1$
- $\bar{u}_2(B) = \max\{1, 1\} = 1$

A Worked Example

1 \ 2	L	R
T	-1, -3	1, 0
M	2, -1	0, 1
B	1, 1	-2, 1

- Player 2's punishment strategy against 1 would be to choose a strategy σ_j that minimizes $\bar{u}_1(\sigma_j)$, i.e., R
- Player 1's punishment strategy against 2 would be to choose a strategy σ_j that minimizes $\bar{u}_2(\sigma_j)$, i.e., T

So the security values are $(1, 0)$: every outcome in which the respective players get at least these values can be sustained as an NE.

The outcomes are: (T,R), (B,L).

A Worked Example

Sanity check:

Draw finite state machine strategies for both players to sustain (B,L) as an NE.

Discounted Sum

A key technique for computing utility of infinite runs

- Idea is to use a **discount factor**, $0 < \delta < 1$, to discount the value of future rounds
- The value of the infinite run

$$\omega_0 \quad \omega_1 \quad \omega_2 \quad \cdots$$

to player *i* is then

$$\sum_{u \in \mathbb{N}} \delta^u u_i(\omega_u) = u_i(\omega_0) + \delta u_i(\omega_1) + \delta^2 u_i(\omega_2) + \cdots$$

- The core identity for computing discounted sums is:

$$\sum_{n=1}^{\infty} \delta^n = \frac{1}{1-\delta}$$

Part XXX

Iterated Boolean Games

Iterated Boolean Games (iBG)

- A model of multi-agent systems in which players repeatedly choose truth values for Boolean variables under their control.
- Players behave selfishly in order to achieve individual **goals**.
- Goals expressed as **Linear Temporal Logic** (LTL) formulae.

Propositional Linear Temporal Logic (LTL)

A standard language for talking about **infinite state sequences**.

$\top$	truth constant
p	primitive propositions ($\in \Phi$)
$\neg\varphi$	classical negation
$\varphi \vee \psi$	classical disjunction
$\mathbf{X}\,\varphi$	in the next state…
$\mathbf{F}\,\varphi$	will eventually be the case that φ
$\mathbf{G}\,\varphi$	is always the case that φ
$\varphi\,\mathbf{U}\,\psi$	φ until ψ

Example LTL formulae

$$\mathbf{F}\,\neg \mathit{pandemic}$$

eventually there will not be a pandemic (a **liveness** property)

$$\mathbf{G}\,\neg \mathit{crash}$$

the plane will never crash (a **safety** property)

$$\mathbf{G}\,\mathbf{F}\,\mathit{drinkBeer}$$

I will drink beer **infinitely often**

$$\mathbf{F}\,\mathbf{G}\,\mathit{dead}$$

eventually will come a time at which I am dead forever after.

$$(\neg \mathit{friends})\,\mathbf{U}\,\mathit{youApologise}$$

Example LTL formulae

$$\mathbf{F}\,\neg \mathit{pandemic}$$

eventually there will not be a pandemic (a **liveness** property)

$$\mathbf{G}\,\neg \mathit{crash}$$

the plane will never crash (a **safety** property)

$$\mathbf{G}\,\mathbf{F}\,\mathit{drinkBeer}$$

I will drink beer **infinitely often**

$$\mathbf{F}\,\mathbf{G}\,\mathit{dead}$$

eventually will come a time at which I am dead forever after.

$$(\neg \mathit{friends})\,\mathbf{U}\,\mathit{youApologise}$$

we are not friends **until** you apologise

Iterated Boolean games

An iBG is a structure

$$G = (N, \Phi, \Phi_1, \ldots, \Phi_n, \gamma_1, \ldots, \gamma_n)$$

where

- $N = \{1, \ldots, n\}$ is a set of **agents** (the players of the game),
- $\Phi = \{p, q, \ldots\}$ is a finite set of **Boolean variables**,
- $\Phi_i \subseteq \Phi$ is the set of variables controlled by player i,
- γ_i is the **LTL goal** of player i.

Models for LTL

- Let V be the set of **valuations** of Boolean variables Φ.
- Let V_i be the valuations for the variables Φ_i controlled by player i.
- Models of LTL formulae φ are **runs** ρ: infinite sequences in V^ω.
- We write $\rho \models \varphi$ to mean ρ satisfies LTL formula φ.

Playing an iBG

- Players play an infinite number of rounds, where on each round each player chooses values for their variables.
- The sequence of valuations traced out in this way forms a run, which either satisfies or doesn't satisfy a player's goal.
- A **strategy** for i is thus abstractly a function

$$f : V^* \to V_i$$

  ... but this isn't a **practicable** representation.
- So we model strategies as **finite state machines (FSM) with output** (transducers).

Machine strategies

A machine strategy for i is a structure:

$$\sigma_i = (Q_i, q_i^0, \delta_i, \tau_i)$$

where:

- Q_i is a finite, non-empty set of **states**,
- q_i^0 is the **initial** state,
- $\delta_i : Q_i \times V \rightarrow Q_i$ is a **state transition function**,
- $\tau_i : Q_i \rightarrow V_i$ is a **choice function.**

Strategy profiles

- A **strategy profile** $\vec{\sigma}$ is an n-tuple of machine strategies, one for each player i:
$$\vec{\sigma} = (\sigma_1, \ldots, \sigma_n).$$
- As strategies are **deterministic**, each strategy profile $\vec{\sigma}$ induces a unique run: $\rho(\vec{\sigma})$.

Nash Equilibrium

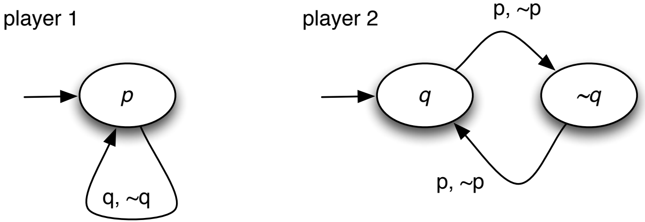
Strategy profile $\vec{\sigma} = (\sigma_1, \ldots, \sigma_i, \ldots, \sigma_n)$ is a (pure strategy) **Nash equilibrium** if for all players $i \in N$, if $\rho(\vec{\sigma}) \not\models \gamma_i$ then for all σ'_i we have

$$\rho(\sigma_1, \ldots, \sigma'_i, \ldots, \sigma_n) \not\models \gamma_i$$

Let $NE(G)$ denote the Nash equilibria of a given iBG G.

An Example

- $N = \{1, 2\}$,
- $\Phi_1 = \{p\}$
- $\Phi_2 = \{q\}$
- $\gamma_1 = \mathbf{G}\,\mathbf{F}(p \leftrightarrow q)$
- $\gamma_2 = \mathbf{G}\,\mathbf{F}\,\neg(p \leftrightarrow q)$



These strategies form a NE.

Decision problems

MODEL CHECKING:
Given: Game G, strategy profile $\vec{\sigma}$, and LTL formula φ.
Question: Is it the case that $\rho(\vec{\sigma}) \models \varphi$?

MEMBERSHIP:
Given: Game G, strategy profile $\vec{\sigma}$.
Question: Is it the case that $\vec{\sigma} \in \mathit{NE}(G)$?

Decision problems

MODEL CHECKING:
Given: Game G, strategy profile $\vec{\sigma}$, and LTL formula φ.
Question: Is it the case that $\rho(\vec{\sigma}) \models \varphi$?

MEMBERSHIP:
Given: Game G, strategy profile $\vec{\sigma}$.
Question: Is it the case that $\vec{\sigma} \in NE(G)$?

Theorem

The MODEL CHECKING *and* MEMBERSHIP *problems are PSPACE-complete.*

Proof: follow from the fact that we can encode FSM strategies as LTL formulae.

Decision problems

E-NASH:
Given: Game G, LTL formula φ.
Question:$\exists \vec{\sigma} \in NE(G).\ \rho(\vec{\sigma}) \models \varphi$?

A-NASH:
Given: Game G, LTL formula φ.
Question: $\forall \vec{\sigma} \in NE(G).\ \rho(\vec{\sigma}) \models \varphi$?

NON-EMPTINESS:
Given: Game G.
Question: Is it the case that $NE(G) \neq \emptyset$?

Decision problems

E-NASH:
Given: Game G, LTL formula φ.
Question:$\exists \vec{\sigma} \in NE(G).\ \rho(\vec{\sigma}) \models \varphi$?

A-NASH:
Given: Game G, LTL formula φ.
Question: $\forall \vec{\sigma} \in NE(G).\ \rho(\vec{\sigma}) \models \varphi$?

NON-EMPTINESS:
Given: Game G.
Question: Is it the case that $NE(G) \neq \emptyset$?

Theorem

The E-NASH, A-NASH, *and* NON-EMPTINESS *problems are 2EXPTIME-complete.*

Proof: we can reduce **LTL synthesis** (Pnueli & Rosner, 1989)

Folk theorems in iBGs

- For iBGs, the Folk (Nash) Theorems for iBG answer the question:

  Which LTL properties are satisfied in the Nash equilibria of a given iterated Boolean game?

- In other words, which LTL formulae will be true if everyone acts rationally?

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

  I'll cooperate if she will.
- **Program equilibria**[9] 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.**

---

[9] 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.

Part XXXI

Evolutionary Games

Evolutionary 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[10].

[10] 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.

Why did TIT-FOR-TAT do well?

Perhaps surprising that TIT-FOR-TAT do so well...

Proposition

*In all 2 player finitely repeated prisoner's dilemma games, TIT-FOR-TAT does no better (and possibly worse) than **all** other strategies: in a one-to-one competition, it does no better than any possible strategy.*

So what is the explanation?

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.

This is not mathematically robust advice – somewhat controversial amongst game theorists.

So, why does TIT-FOR-TAT do so well?

- If TIT-FOR-TAT was in a population of ALLD, it would suffer.
- **But it isn't.** It is in a population that contains **cooperative** agents
- TIT-FOR-TAT does well because it **gets to play against other cooperative strategies**: the "strategy population" consisted of other cooperative strategies.
- When cooperative strategies meet, they can **share** the benefits of mutual cooperation, while strategies that immediately defect can get bogged down in conflict

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)

Evolutionary game theory[11]

- For Axelrod, the exciting thing was that TIT-FOR-TAT, and mutually sustained cooperation, could arise merely through blind evolutionary processes: **cooperation through evolution.**
- There is no “thinking” about what strategy to choose
- Strategies are chosen through natural selection
- In **evolutionary game theory**, we have:

$$\text{evolutionary fitness} = \text{utility}$$

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

The Hawk-Dove Game

- Suppose have a very large population of individuals, which come in two variants: **Hawks** and **Doves**
- These variants play role of strategies in conventional game theory.
- Individual don't "decide" whether to be a Hawk or a Dove: variants are **genetically hardwired**
- Individuals reproduce over time, but reproduction is **asexual**: individual doesn't need a partner to reproduce, and if an individual reproduces, it begets offspring of the same type.
- The key attribute of an individual that determines how likely they are to reproduce is a numeric value that we'll call their fitness, which measures **how likely that individual is to be able to reproduce and pass on their genes**

The Hawk-Dove Game

- In the Hawk-Dove Game, individuals increase their fitness by obtaining a particular resource (e.g., food) from the environment.
- Individuals are in competition with others to obtain resources.
- Hawks are fierce; Doves are timid. . .

1. When a Hawk competes with a Dove, the Hawk takes the whole of the resource.
2. When a Dove competes with a Dove, they share the resource equally.
3. When a Hawk competes with a Hawk, they fight, and have an equal chance of obtaining the resource or being injured.

The Hawk-Dove Game

- V denotes the value of the resource
  This is the **increase in fitness** that an individual would gain by obtaining the resource.

- C denotes the **cost of injury**
  This is the amount by which fitness would **decrease** if an individual fought for the resource and lost.

Rules of the Hawk-Dove Game

1. When a Hawk meets a Hawk: they fight, and have an equal chance of increasing their fitness by V or decreasing their ftness by C; on average, this will result in an increase of fitness by $(V - C)/2$.
2. When a Dove meets a Dove, they share the resource equally, each obtaining an increase of ftness of $V/2$.
3. When a Hawk meets a Dove, the Hawk takes the whole of the resource, giving V to the Hawk, while the Dove gets no benefit.

Fitness

- Let p denote **proportion of Hawks** in population.
- Let $W(H)$ and $W(D)$ denote **average fitness** of Hawks and Doves:

$$
\begin{aligned}
W(H) &= p((V - C)/2) + (1 - p)V \\
W(D) &= (1 - p)(V/2)
\end{aligned}
$$

- The **expected utility** of playing the Hawk-Dove game.

Replicator Dynamics

Describe how populations change over time

- Let A denote the **average fitness of the population**:

$$A = pW(H) + (1 - p)W(D)$$

  i.e., the expected fitness of an individual drawn uniformly at random from the population

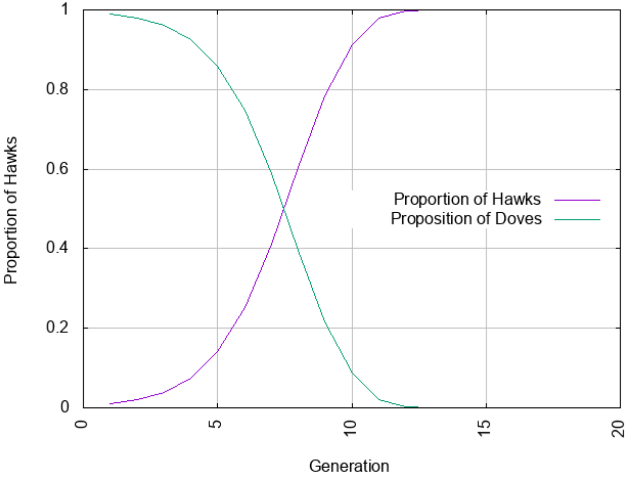
- The new frequency p' of Hawks in the next generation is then:

$$p' = p\frac{W(H)}{A}$$

- Thus if $W(H) > A$ then Hawks will **increase** in next generation, those whose fitness is **less** will decrease

Replicator Dynamics in the Hawk-Dove Game

$V = 4, C = 3, P = 0.01$

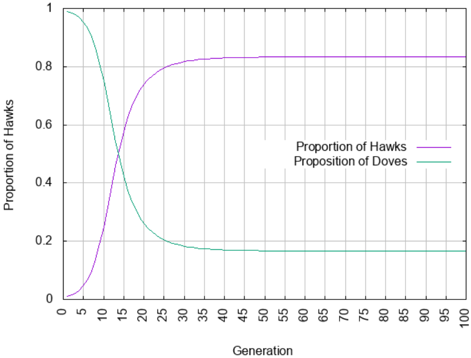


Replicator Dynamics in the Hawk-Dove Game

Suppose, however, the payoffs are as follows...

	Hawk	Dove
Hawk	0.4	1.5
Dove	0.5	1.0

...then Doves & Hawks can coexist!



Evolutionarily stable strategies (ESS)

Formally, strategy σ is an ESS iff:

1. It is a best response to itself.
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.)

Computational Game Theory

Lecture 6: Normal Form Zero Sum Games

UNIVERSITY OF OXFORD
DEPARTMENT OF COMPUTER SCIENCE

Zero Sum Games

- Recall that zero sum games are games in which for every outcome $\omega \in \Omega$ we have

$$\sum_{i \in N} u_i(\omega) = 0$$

- Zero sum games are **strictly competitive**:

☞ **best outcome for me** is **worst outcome for you** ☜

- The **symmetry** of preferences in zero sum games means we can use different (simpler!) techniques to analyse them.

☞ **Zero sum games are different!** ☜

How should you play a zero sum game?

Only list utilities of row player in zero sum payoff matrices.

	X	Y	Z
A	2	5	13
B	6	5.6	10.5
C	6	4.5	1
D	10	3	-2

Imagine you are the row player and you make a choice first; column player then gets to respond.

How should you play a zero sum game?

Only list utilities of row player in zero sum payoff matrices.

	X	Y	Z
A	2	5	13
B	6	5.6	10.5
C	6	4.5	1
D	10	3	-2

Imagine you are the row player and you make a choice first; column player then gets to respond. You know that whichever row you choose, column player will pick your smallest utility in that row.

How should you play a zero sum game?

Only list utilities of row player in zero sum payoff matrices.

	X	Y	Z
A	2	5	13
B	6	5.6	10.5
C	6	4.5	1
D	10	3	-2

Imagine you are the row player and you make a choice first; column player then gets to respond. You know that whichever row you choose, column player will pick your smallest utility in that row. $\Rightarrow$ You should choose the row that **maximises** that **minimum**.

The Maximin Value of a Zero Sum Game

	X	Y	Z	min
A	2	5	13	2
B	6	5.6	10.5	5.6
C	6	4.5	1	1
D	10	3	-2	-2

Take the minimum of each row.

The Maximin Value of a Zero Sum Game

	X	Y	Z	min
A	2	5	13	2
B	6	5.6	10.5	5.6
C	6	4.5	1	1
D	10	3	-2	-2

Take the minimum of each row.

The **maximin value**, $\bar{v}$, is then the maximum of these.

The Maximin Value of a Zero Sum Game

	X	Y	Z	min
A	2	5	13	2
B	6	5.6	10.5	5.6
C	6	4.5	1	1
D	10	3	-2	-2

Take the minimum of each row.
The **maximin value**, $\bar{v}$, is then the maximum of these.

$$\bar{v} = \max_{\sigma_1 \in \Sigma_1} \min_{\sigma_2 \in \Sigma_2} u_1(\sigma_1, \sigma_2) = 5.6$$

Safety Strategies

But remember: life is Not a zero-sum game!

- Strategies that yield maximin outcomes are sometimes called **safety strategies**
- The maximin value for a player is sometimes called the **safety level** or **reservation value**
- This is “worst case scenario” thinking
- Is it ever really justified?
  - zero-sum games are rare
  - maybe makes sense against **irrational** players. . . ?
- Unfortunately, in practice, people often interact as though in a zero-sum game, and miss out on benefits of common interest

The Minimax Value of a Zero Sum Game

	X	Y	Z	min
A	2	5	13	2
B	6	5.6	10.5	5.6
C	6	4.5	1	1
D	10	3	-2	-2
max	10	5.6	13	

Take the maximum of each column.

The Minimax Value of a Zero Sum Game

	X	Y	Z	min
A	2	5	13	2
B	6	5.6	10.5	5.6
C	6	4.5	1	1
D	10	3	-2	-2
max	10	5.6	13	

Take the maximum of each column. The **minimax value**, $\underline{v}$, is then the minimum of these.

The Minimax Value of a Zero Sum Game

	X	Y	Z	min
A	2	5	13	2
B	6	5.6	10.5	5.6
C	6	4.5	1	1
D	10	3	-2	-2
max	10	5.6	13	

Take the maximum of each column. The **minimax value**, $\underline{v}$, is then the minimum of these.

$$\underline{v} = \min_{\sigma_2 \in \Sigma_2} \max_{\sigma_1 \in \Sigma_1} u_1(\sigma_1, \sigma_2)$$

The Minimax Theorem (Pure Strategies)

Theorem (John von Neumann, 1928.)

Suppose we have a two player zero-sum game, in which (σ_1, σ_2) is a Nash equilibrium. Then:

$$u_1(\sigma_1, \sigma_2) = \bar{v} = \underline{v}$$

Thus, in zero-sum games, Nash equilbria and maximin/minimax outcomes coincide, and a player receives the same utility in each Nash equilibrium.

The maximin value for player 1 is called the **value of the game.** (Player 2 gets $-\bar{v}$.)
(Von Neumann proved it for mixed strategies; a much more complex and significant result.)

Computing Mixed Strategies in Zero-Sum Games

- The symmetry in two player zero-sum games means that computing mixed NE is an **optimization problem**
- This optimization problem can be solved via **linear programming**

Minimax Linear Program for Player 2

minimize U_1^* subject to:

$$\sum_{\sigma_2^k \in \Sigma_2} u_1(\sigma_1, \sigma_2^k) \cdot p_2^k \leq U_1^* \qquad \text{for all } \sigma_1 \in \Sigma_1$$

$$\sum_{\sigma_2^k \in \Sigma_2} p_2^k = 1$$

$$p_2^k \geq 0 \qquad \text{for all } \sigma_2^k \in \Sigma_2$$

Here the unknowns are $U_1^*, p_2^1, \ldots, p_2^l$
Values p_2^k give probability of player 2 choosing σ_2^k

Maximin Linear Program for Player 1

maximize U_1^* subject to:

$$\sum_{\sigma_1^j \in \Sigma_1} u_1(\sigma_1^j, \sigma_2) \cdot p_1^j \geq U_1^* \qquad \text{for all } \sigma_2 \in \Sigma_2$$

$$\sum_{\sigma_1^j \in \Sigma_1} p_1^j = 1$$

$$p_1^j \geq 0 \qquad \text{for all } \sigma_1^j \in \Sigma_1$$

Values p_1^j give probability of player 1 choosing σ_1^j

Computational Game Theory

Lecture 6: Extensive Form Win-Lose Games

UNIVERSITY OF OXFORD

DEPARTMENT OF COMPUTER SCIENCE

Extensive Form Win-Lose Games

- We focus on 2 player games, with players E and A.
- We are interested in two-player zero-sum games (i.e., strictly competitive games).
- Assume leaf nodes are labelled with either 1 or -1, indicating payoff for player E:
  - payoff = 1 means “player E wins”
  - payoff = -1 means “player E loses”

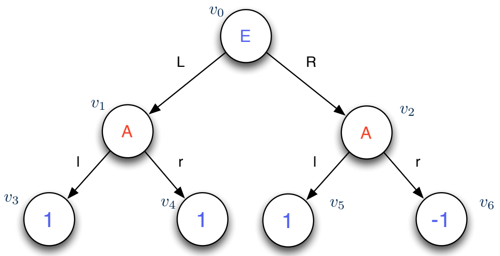
Determinacy

- Key concept in win-lose games is whether games are **determined**: whether some player can **force a win.**
- You forcing a win means that you have a strategy such that all outcomes possible by playing that are strategy result in a win for you:

there exists a choice for you
 such that for all choices of your counterpart
  there exists a choice for you
   such that for all choices of your counterpart
    . . .
     you win.

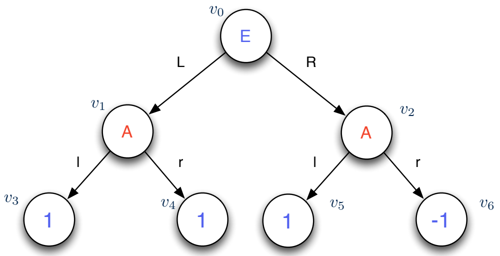
☞ Hence we talk of **winning strategies** for players. ☜

Example Extensive Form Win-Lose Game



As usual in zero-sum games, we only list payoffs for one player, in this case the blue player.

Example Extensive Form Win-Lose Game



Zermelo's algorithm works fine for such games: player *E* can force a win ($=$payoff 1) by choosing *L*.

Example Extensive Form Win-Lose Game

Theorem

In extensive form win-lose games, Zermelo's algorithm tells us which player can force a win; the strategies computed by Zermelo's algorithm are the optimal strategies for all players, and in particular, give us a winning strategy for the relevant player.

As a corollary, (finite) extensive form win-lose games are determined: one of the players can force a win.

Example Extensive Form Win-Lose Game

Theorem

In extensive form win-lose games, Zermelo's algorithm tells us which player can force a win; the strategies computed by Zermelo's algorithm are the optimal strategies for all players, and in particular, give us a winning strategy for the relevant player.

As a corollary, (finite) extensive form win-lose games are determined: one of the players can force a win.

Theorem

Determining whether a given player has a winning strategy in a finite win-lose extensive form game is P-complete.

Extensive Form Win-Lose Games in Computer Science

- Two-player extensive form win-lose games have a particularly important role in computer science, although game-theoretically, they are quite simple.
- Many computer science decision problems can be formulated as extensive-form win-lose games.
- Typical formulation:

  x is a positive instance of the decision problem Π iff player 1 can force a win in the game $G_{(\Pi,x)}$.

- Different types of extensive form win-lose games characterise different complexity classes (P, PSPACE, EXPTIME)

Compactly Represented Games

- Zermelo's algorithm finds winning strategies in polynomial time
- However, assumes that **the entire game tree is given as an input**: running time is polynomial **in the size of the tree**.
- In many cases, we work with **compact** representations of game trees.
- Where the game tree is represented in a compact way, we might expect the complexity to increase… and it does.

Complexity of Win-Lose Extensive Form Games

1. Explicitly represented games (game tree in input)
   ☞ typically **P-complete**
2. Games guaranteed to end after a small number of moves
   ☞ typically **PSPACE-complete**
3. Games that can go on for exponentially many moves
   ☞ typically **EXPTIME-complete**
4. Games that can go on for a long time **and** require memory.
   ☞ typically **EXPSPACE-complete**
5. Games that go on for ever
   ☞ high complexity!

Games with a Small Number of Moves

The Formula Game

- The formula game is played by two players, and is defined by variables

$$\vec{x} = x_1, \ldots, x_k$$

$$\vec{y} = y_1, \ldots, y_k$$

and propositional logic formula

$$\varphi(\vec{x}, \vec{y}).$$

- Player one picks a value ($\top$ or $\bot$) for x_1, then 2 picks a value for y_1, and so on, until all variables have a value.
- Player 1 wins if φ is made true under the valuation they define in this way.

The Formula Game

- The formula game is a **compactly specified extensive form game, with a small number of moves** (*k* moves for each player).
- We can solve it by unfolding the corresponding game tree and applying Zermelo…
- … but the game tree will be of size exponential in the number of variables.

The Formula Game

Notice that game is a win for player 1 iff:

there exists a value for x_1
such that for all values of y_1
there exists a value for x_2
such that for values of y_2
$\ldots$
φ *is true under the resulting valuation.*

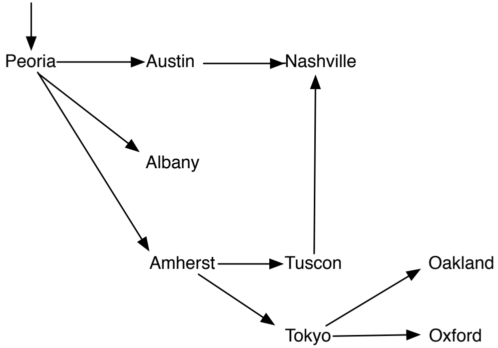
But this is just the **Quantified Boolean Formula (QBF)**:

$$\exists x_1 \, \forall y_1 \, \exists x_2 \, \forall y_2 \, \cdots \, \varphi(\vec{x}, \vec{y}).$$

Theorem

Checking whether player 1 can force a win in the formula game is PSPACE-complete.

The Game of Geography



- Start with player 1 naming the initial city (in this case Peoria)
- Players alternate to name another city whose name starts with the last letter of the previous city name.
- Not allowed to name same city twice.
- Player loses if can't move.

Geography

Theorem

Checking whether player 1 *can force a win in the game of Geography is PSPACE-complete.*

Games in PSPACE

Suppose we have a compactly specified win-lose game G in which:

1. the length of any legal sequence of moves is bounded by a polynomial in the size of the input; and
2. given a “board position”, there is a polynomial space algorithm that constructs all possible next moves and board positions, or else decides which player has won.

Then G can be solved in PSPACE.

An instance of PEEK-G_4 is a quad:

$$\langle X_1, X_2, X_3, \varphi \rangle \text{ where:}$$

- X_1 and X_2 are disjoint, finite sets of Boolean variables, with the intended interpretation that the variables in X_1 are under the control of agent 1, and X_2 are under the control of agent 2;
- $X_3 \subseteq (X_1 \cup X_2)$ are the variables deemed to be true in the initial state of the game; and
- φ is a propositional logic formula over the variables $X_1 \cup X_2$, representing the winning condition.

Game starts from the initial assignment X_3

Players alternate (1 moving first) to select a value for **one** of their variables (passing allowed)

Player i wins if they make a move resulting in φ being true.

The Game PEEK-G_4

Observation: If a player can win she can do so in $O(2^{|X|})$ moves (why?)

Theorem

Checking whether a given player has a winning strategy in an instance of PEEK-G_4 *is EXPTIME-complete.*

Can you give an algorithm that solves in EXPTIME?

Solving PEEK-G_4

- First idea: unfold into a game tree and apply Zermelo.
- Doesn't work because the tree gets too large:
  - a tree of branching factor b and depth d will have b^d states at the end
  - in this case, $b = |X_i|$, $d = 2^{|X|}$ so $b^d = |X_i|^{2^{|X|}}$
  - this is a **2EXPTIME** algorithm.
- But with a little care we **can** still use backwards induction. . .

Solving PEEK-G_4

- Define a state as a valuation for X together with a label indicating whose turn it is in that state
- We'll have $2^{|X|+1}$ states overall
- Then construct two graphs $\longrightarrow_1$ and $\longrightarrow_1$ on the stateset, where:

  $s_1 \longrightarrow_1 s_2$
  if it is 1's turn to move in s, and there is a move for 1 that transforms s_1 into s_2

  $s_1 \longrightarrow_2 s_2$
  if it is 2's turn to move in s, and there is a move for 2 that transforms s_1 into s_2

Solving PEEK-G_4

- Systematically loop through each state: if the state satisfies φ and it is player i's move, label the state as a win for j. (These are the "leaf nodes".)
- Iterate over the state set, labelling a state as a win for player i if either:
  - it is i's turn to move and i has a choice that leads to a state that is labelled as a win for i
  - it is j's turn to move and all choices that j can make lead to states that are labelled as a win for i.

  Repeat until no changes.

Theorem

Player i has winning strategy iff the initial state is labelled as a win for player i.

EXPSPACE Games...

- Consider an instance of PEEK-G_4 where **you are not allowed to revisit the same valuation of variables**
- Here, you have to **remember all previous board configurations**.
- There are no shortcuts here: you have to store them all, and so...

Theorem

PEEK-G_4 *without repetitions is EXPSPACE-complete.*