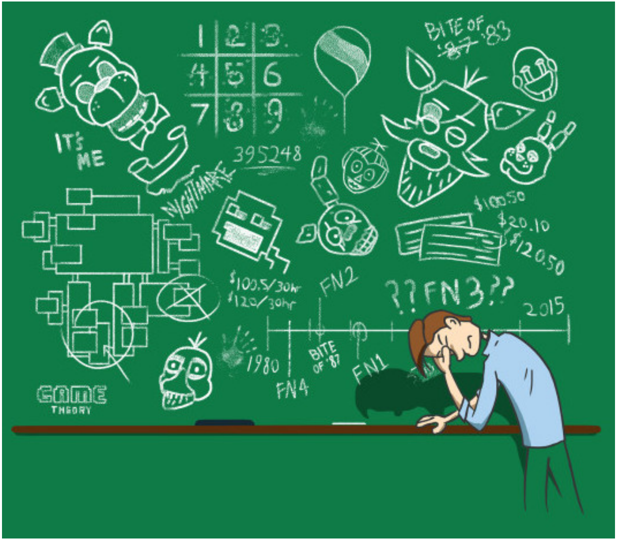
Topic 2: Basic Models



Outcomes & Objectives

- Be proficient in modeling games mathematically
    - Apply decision-theoretic concepts (e.g. lotteries, utilities) to model agent decisions and outcomes in a game.
    - Use mathematical structures (e.g. matrices, graphs) to represent the state of the game.
    - Transform from one representation to another (e.g. extensive-form to normal-form and vice versa).
    - Identify some useful properties in games (e.g. zero-sum games, games with information asymmetry).

- Be proficient with basic solution approaches.
    - Iterative Elimination of Dominated Strategies
    - Minimax Equilibrium
    - Nash Equilibrium

- Apply game theory in various applications.
    - Congestion games in transportation
    - MAC-layer games in computer/wireless networks
    - Game-theoretic security

Games: Types and Representations

Definition

Game is a strategic framework where multiple intelligent agents interact with one another through their rational decisions.

Types of games:

- Non-cooperation vs. Cooperation
- Static vs. Dynamic
- Perfect-information vs. imperfect-information
- Complete-information vs. incomplete-information

Two basic representations:

- ***Normal/Strategic Form***: Matrix Representation
- ***Extensive Form***: Graph (Decision-Tree) Representation

Normal-Form Representation

Definition

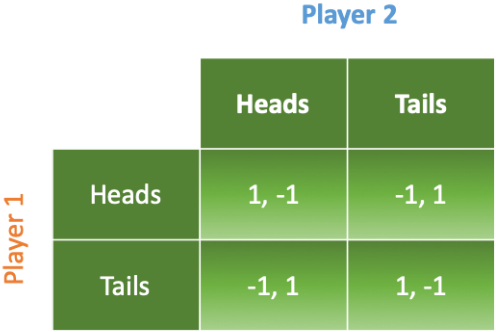
A ***normal-form (or a strategic-form) game*** Γ is defined as a triplet $(\mathcal{N}, \mathcal{C}, \mathcal{U})$, where

- $\mathcal{N} = \{1, \cdots, N\}$ is the set of N players (agents),
- $\mathcal{C} = \mathcal{C}_1 \times \cdots \times \mathcal{C}_N$ is the strategy profile space, where $\mathcal{C}_i$ represents the set of strategic choices (actions) available at the i^{th} player,
- $\mathcal{U} = \{u_1, \cdots, u_N\}$ is the set of utility functions, where $u_i : \mathcal{C}_i \to \mathbb{R}$ represents the utility function at the i^{th} player.

Example: Matching Pennies

Two players toss their respective coins and compare their outcomes.

- $\mathcal{N} = \{1, 2\}$ (Two-player game),
- $\mathcal{C} = \{H, T\} \times \{H, T\}$,
- $\mathcal{U} = \{u_1, u_2\}$, where $u_i : \mathcal{C}_i \to \{-1, 1\}$ such that $u_1 + u_2 = 0$.



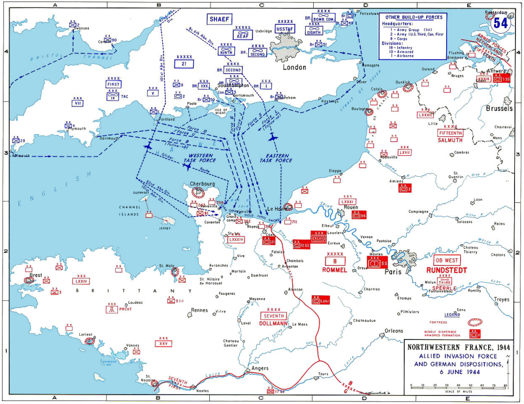
Player 2

		Heads	Tails
Player 1	Heads	1, -1	-1, 1
	Tails	-1, 1	1, -1

Matching Pennies: Applications

- **Sports:** Soccer penalty kicks, Tennis serve-and-return plays
- **Security:** Attack-defense games in computer security, cops vs. adversaries in airports

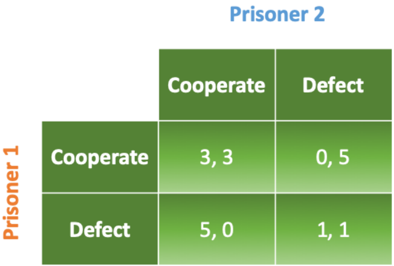
Allied landing in Europe on June 6, 1944: Normandy vs. Calais



Example: Prisoner's Dilemma

Two prisoners involved in the same crime are being interrogated simultaneously in separate rooms. They can either cooperate or defect with the interrogators.

- $\mathcal{N} = \{P_1, P_2\}$
- $\mathcal{C} = \{C, D\} \times \{C, D\}$
- $\mathcal{U} = \{u_1, u_2\}$, where $u_i : \mathcal{C}_i \rightarrow \mathbb{R}$, as shown in the matrix below.



		Prisoner 2	
		Cooperate	Defect
Prisoner 1	Cooperate	3, 3	0, 5
	Defect	5, 0	1, 1

Prisoner's Dilemma: Applications

- **Networking:** *CSMA with Collision Avoidance* (a.k.a. TCP User's Game)
- **Climate Change Politics:** No country is motivated to curb CO_2 emissions for selfish reasons, although every country benefits from a stable climate.
- **Advertising:** Two competiting firms can either advertise, or not advertise about their products at a given time.
- **Peer-to-Peer File Sharing:** BitTorrent's *unchoking* strategies in search of cooperative peers to optimize downlink data-rates resemble those in this game.

Captures lack of trust between players!

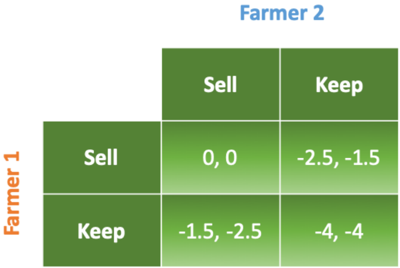
Will study in detail in Topic 3 (Advanced Solution Concepts).

Example: Tragedy of the Commons

- $\mathcal{N} = \{F_1, \cdots, F_n\}$
- Farmer i (F_i): Keep the sheep or not ($s_i \in \{0, 1\}$)
- Payoff for keeping the sheep = 1 unit
- Village has limited stretch of grassland
- Damage to environment = 5 units (shared equally by all farmers)

Net utility at F_i: $$u_i(s_1, \cdots, s_n) = s_i - 5\left[\frac{s_1 + \cdots + s_n}{n}\right]$$

If $n = 2$:

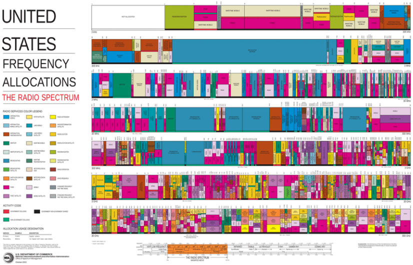


		Farmer 2	
		Sell	Keep
Farmer 1	Sell	0, 0	-2.5, -1.5
	Keep	-1.5, -2.5	-4, -4

Tragedy of the Commons: Applications

Application: Spectrum Commons

- 3650 MHz (50 MHz block): Licensed Commons
- Wifi (2.4 GHz, 5 GHz): Unlicensed Commons

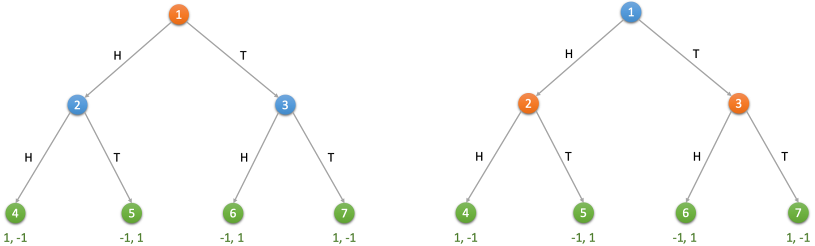


A multi-player generalization of Prisoner's Dilemma!

Extensive-form representation captures more information!

- state evolution in a game and the corresponding choice sets
- order of moves
- information available throughout the game

Play-Order in Matching Pennies:



Observability: Perfect vs. Imperfect Information

Definition

A game where every agent can observe every other player's actions is called a ***perfect information game***.

Example: Chess

Imperfect Information: Player's actions are not observable!
Example: Poker

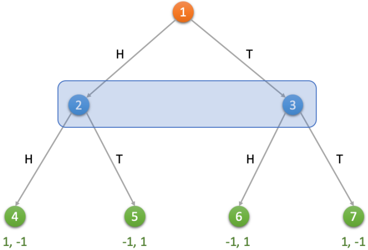
Games which are sequential, and which have chance events, but no secret information, are considered games of perfect information.

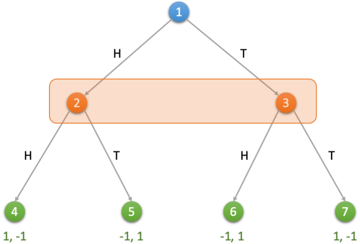
Example: Monopoly (uncertainty due to rolling dice.)

More on Imperfect Information Games...

Games with simultaneous moves are generally considered imperfect information games!

Matching Pennies with Simultaneous Moves:





Group all indistinguishable states into sets to disclose available information at each agent!

Nature’s Role in Games

- Players play the left game with probability p,
- Players play the right game with probability $1 - p$,

Player 2

Player 1

	Heads	Tails
Heads	1, -1	-1, 1
Tails	-1, 1	1, -1

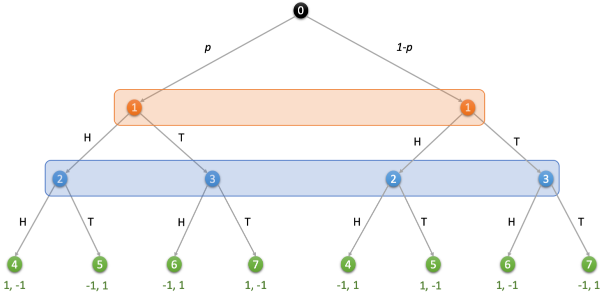
Left Game (*p*)

Player 2

Player 1

	Heads	Tails
Heads	-1, 1	1, -1
Tails	1, -1	-1, 1

Right Game (1 - *p*)



Agent Types: Complete vs. Incomplete Information

Sometimes, players may not know each others' types.

Such games are called incomplete-information (or Bayesian) games.

Definition

A ***Bayesian (or incomplete information game) game*** Γ is defined as a tuple $(\mathcal{N}, \Theta, \boldsymbol{p}, \mathcal{C}, \mathcal{U})$, where

- $\mathcal{N} = \{1, \cdots, N\}$ is the set of N players (agents),
- $\Theta = \{\Theta_1, \cdots, \Theta_N\}$, where Θ_i is the set of types of player i,
- $\boldsymbol{p} = \{p_1, \cdots, p_N\}$, where $p_i : \Theta_i \rightarrow \Delta(\Theta_{-i})$ is the conditional belief over the set of types of other players, given the type of player i,
- $\mathcal{C} = \mathcal{C}_1 \times \cdots \times \mathcal{C}_N$ is the strategy profile space, where $\mathcal{C}_i$ represents the set of strategic choices (actions) available at the i^{th} player,
- $\mathcal{U} = \{u_1, \cdots, u_N\}$ is the set of utility functions, where $u_i : \mathcal{C}_i \rightarrow \mathbb{R}$ represents the utility function at the i^{th} player.

Example: Competition in Job Markets

Information Sets

- Imperfect observations, nature's randomness and incomplete information about the players' types

  $\Rightarrow$ State uncertainty.
- State uncertainty $\Rightarrow$ Limited information at the agent.

Definition

An ***information set*** $\mathcal{I}_i$ of the i^{th} player P_i is the set of that decision nodes at P_i that are indistinguishable to P_i itself.

Extensive-Form Games: Formal Definition

Definition

An ***extensive-form game*** Γ is defined as a tuple $\Gamma = (\mathcal{N}, \mathcal{C}, G, \pi, P, \mathcal{I}, \mathcal{U})$, where

- $\mathcal{N} = \{1, \cdots, N\}$ is the set of N players (agents),
- $\mathcal{C} = \mathcal{C}_1 \times \cdots \times \mathcal{C}_N$ is the strategy profile space,
- G is a decision tree rooted at node 0 (chance node) with vertices representing the game's states and edges representing different player decisions,
- π represents the chance probabilities at all the alternatives available at the chance node,
- $P : \tilde{G} \to \mathcal{N}$ represents the player function that associates each proper subhistory $\tilde{G} \in G$ to a certain player,
- $\mathcal{I} = \{\mathcal{I}_1, \cdots, \mathcal{I}_N\}$ represents the set of information sets at all the players,
- $\mathcal{U} = \{u_1, \cdots, u_N\}$ is the set of utility functions.

Equivalence of Representations

Can we eliminate temporal dynamics in extensive-form games to gain substantial conceptual simplification, if questions of timing are inessential to our analysis?

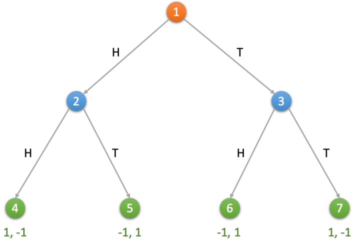
Note: This is not straightforward, i.e.,

$$\Gamma_e = (\mathcal{N}, \mathcal{C}, G, \pi, P, \mathcal{I}, \mathcal{U}) \not\Rightarrow \Gamma = (\mathcal{N}, \mathcal{C}, \mathcal{U})$$

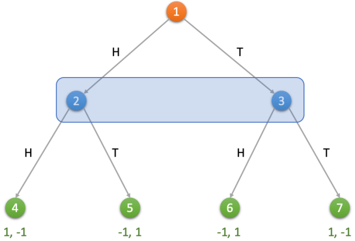
due to the presence of information sets $\mathcal{I}$, play-order, and nature's randomness in π.

Equivalence of Representations (cont...)

Example: Consider the following two Matching Pennies games with non-identical information sets...



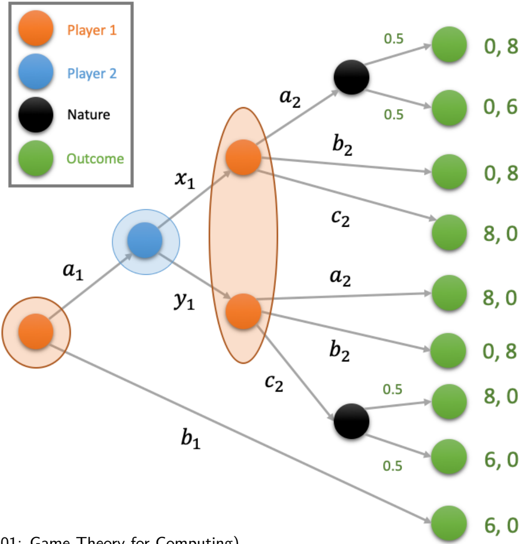
		Player 2			
		(H, H)	(H, T)	(T, H)	(T, T)
Player 1	H	1, -1	1, -1	-1, 1	-1, 1
	T	-1, 1	-1, 1	1, -1	1, -1



		Player 2	
		Heads	Tails
Player 1	Heads	1, -1	-1, 1
	Tails	-1, 1	1, -1

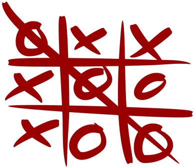
Equivalence of Representations (cont...)

Exercise: Transform the following extensive-form game into a normal-form representation:



Transformation in Large Games is Difficult!

Example: **Tic-Tac-Toe**

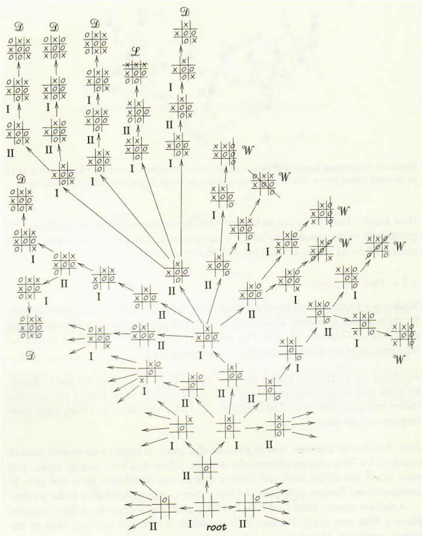


- $\mathcal{N} = \{1, 2\}$
- Environment: 3×3 grid
- Player 1: Place a *cross* (x) in a blank space
- Player 2: Place a *nought* ($\circ$) in a blank space
- Possible outcomes: Win, Loose, Draw
- The first player to have three symbols in straight line wins. The other player looses.

Natural to represent in extensive-form...

How about normal-form representation?

Extensive-form representation[a]:



---

[a] Source: K. Binmore, "Playing for Real: A Text on Game Theory," Oxford University Press, 2007.

Solution Concepts for Normal-Form Games

Assume that we can always transform an extensive-form game into a normal-form equivalent.

Specifically, we will focus on the following solution concepts:

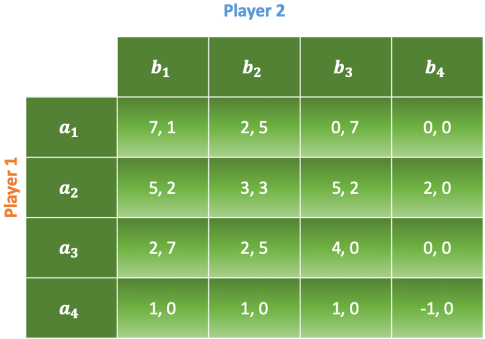
- Iterative Elimination of Dominated Strategies
- Minimax Equilibrium
- Nash Equilibrium

Iterative Elimination of Dominated Strategies

Can we use the notion of dominance to solve games?

Idea: Eliminate one or more dominated strategies at each player in an iterative manner...

Consider the following game:



	b_1	b_2	b_3	b_4
a_1	7, 1	2, 5	0, 7	0, 0
a_2	5, 2	3, 3	5, 2	2, 0
a_3	2, 7	2, 5	4, 0	0, 0
a_4	1, 0	1, 0	1, 0	-1, 0

Iterative Elimination of Dominated Strategies (cont...)

Step 1: $a_3 \succsim a_4 \Rightarrow$ Eliminate a_4

Step 2: $b_3 \succsim b_4 \Rightarrow$ Eliminate b_4

Step 3: $a_2 \succsim a_3 \Rightarrow$ Eliminate a_3

Step 4: $b_2 \succsim b_1 \Rightarrow$ Eliminate b_1

Step 5: $a_2 \succsim a_1 \Rightarrow$ Eliminate a_1

Step 6: $b_2 \succsim b_3 \Rightarrow$ Eliminate b_3

Player 2

Player 1

	~~b_1~~	b_2	~~b_3~~	~~b_4~~
~~a_1~~	~~7, 1~~	~~2, 5~~	~~0, 7~~	~~0, 0~~
a_2	~~5, 2~~	3, 3	~~5, 2~~	~~2, 0~~
~~a_3~~	~~2, 7~~	~~2, 5~~	~~4, 0~~	~~0, 0~~
~~a_4~~	~~1, 0~~	~~1, 0~~	~~1, 0~~	~~1, 0~~

Pure/Mixed Strategies

Definition

Given a choice (strategy) set $\mathcal{C}_i$ at player i, then every $c \in \mathcal{C}_i$ is called a ***pure strategy***.

Definition

Given a player i with a set of pure strategies $\mathcal{C}_i$, a ***mixed strategy*** σ_i is a lottery over $\mathcal{C}_i$.

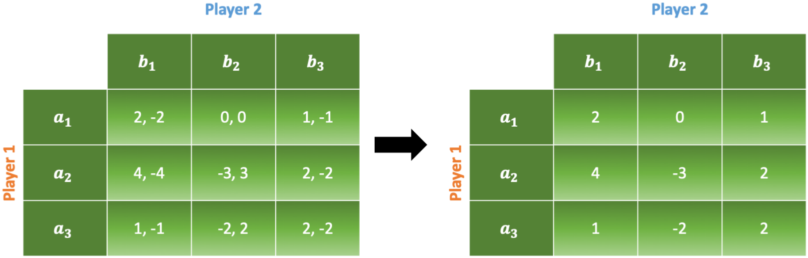
Zero-Sum Games

Definition

A ***zero-sum game*** is the one in which the sum of individual players' utilities for each outcome sum to zero.

Example: Matching Pennies.

In two-player zero-sum games, if Alice (Player 1) wins, Bob (Player 2) looses, and vice versa. Therefore, w.l.o.g, we represent the utility matrix using Alice's utilities.



Player 2

Player 1

	b_1	b_2	b_3
a_1	2, -2	0, 0	1, -1
a_2	4, -4	-3, 3	2, -2
a_3	1, -1	-2, 2	2, -2

Player 2

Player 1

	b_1	b_2	b_3
a_1	2	0	1
a_2	4	-3	2
a_3	1	-2	2

Minimax Equilibrium

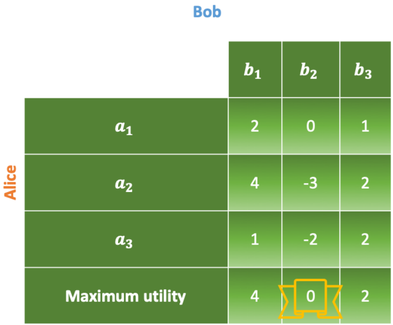
Worst-Case Analysis:

- Alice minimizes her maximum utility (*min-max* strategy).
- Bob maximizes his minimum utility (*max-min* strategy).

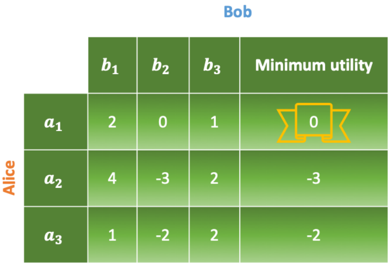
$$\max_{a \in \mathcal{C}_A} \left(\min_{b \in \mathcal{C}_B} u(a,b) \right) \leq u(a,b) \leq \min_{b \in \mathcal{C}_B} \left(\max_{a \in \mathcal{C}_A} u(a,b) \right)$$

Minimax equilibrium is a saddle point in utilities!

Example:



	b_1	b_2	b_3
a_1	2	0	1
a_2	4	-3	2
a_3	1	-2	2
Maximum utility	4	0	2

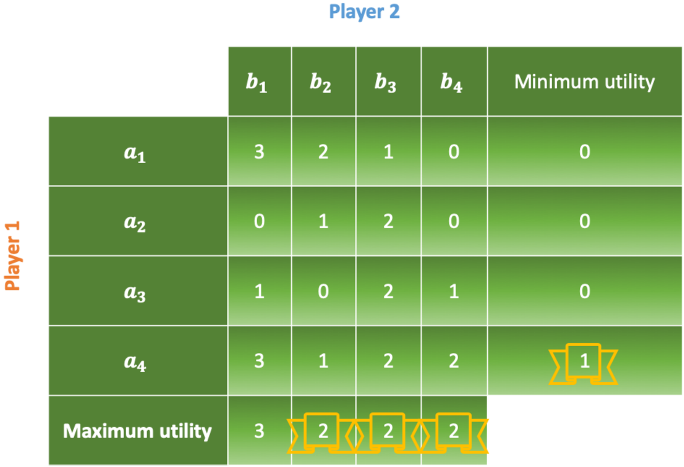


	b_1	b_2	b_3	Minimum utility
a_1	2	0	1	0
a_2	4	-3	2	-3
a_3	1	-2	2	-2

Minimax Equilibrium: (a_1, b_2)

Minimax Equilibrium (cont...)

Example 2:



Player 2

Player 1

	b_1	b_2	b_3	b_4	Minimum utility
a_1	3	2	1	0	0
a_2	0	1	2	0	0
a_3	1	0	2	1	0
a_4	3	1	2	2	1
Maximum utility	3	2	2	2	

Minimax equilibrium may not exist in pure strategies!

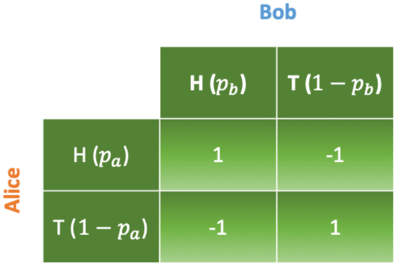
Minimax Equilibrium (cont...)

Minimax equilibrium exists in mixed strategies within finite games!

- Alice minimizes her maximum expected utility (*min-max* strategy).
- Bob maximizes his minimum expected utility (*max-min* strategy).

$$\max_{p_a \in \Delta(\mathcal{C}_A)} \left(\min_{p_b \in \Delta(\mathcal{C}_B)} u(p_a, p_b) \right) \leq u(p_a, p_b) \leq \min_{p_b \in \Delta(\mathcal{C}_B)} \left(\max_{p_a \in \Delta(\mathcal{C}_A)} u(p_a, p_b) \right)$$

Example: Matching Pennies



		Bob	
		H (p_b)	**T ($1-p_b$)**
Alice	**H (p_a)**	1	-1
	T ($1-p_a$)	-1	1

EU: $u(p_a, p_b) = 1 - 2p_a - 2p_b + 4p_a p_b$

Gradiant:

$$\nabla u = \begin{bmatrix} \dfrac{\partial u}{\partial p_a} \\ \dfrac{\partial u}{\partial p_b} \end{bmatrix} = 0 \ \Rightarrow \ p_a = p_b = \frac{1}{2}$$

Hessian matrix: $|\nabla^2 u| < 0 \ \Rightarrow$ Saddle Point!

$$\nabla^2 u = \begin{bmatrix} \dfrac{\partial^2 u}{\partial p_a^2} & \dfrac{\partial^2 u}{\partial p_b \partial p_a} \\ \dfrac{\partial^2 u}{\partial p_a \partial p_b} & \dfrac{\partial^2 u}{\partial p_b^2} \end{bmatrix} = \begin{bmatrix} 0 & 4 \\ 4 & 0 \end{bmatrix}.$$

Best Response of a Player

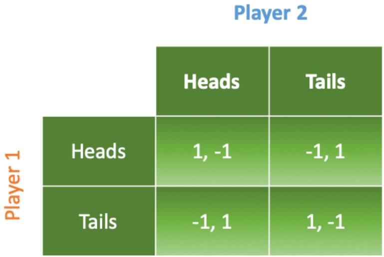
Definition

Given a strategic form game $\Gamma = (\mathcal{N}, \mathcal{C}, \mathcal{U})$ and a strategy profile $\boldsymbol{c}_{-i} \in \mathcal{C}_{-i}$, we say $c_i \in \mathcal{C}_i$ is a ***best response*** of player i with respect to $\boldsymbol{c}_{-i}$ if

$$u_i(c_i, \boldsymbol{c}_{-i}) \geq u_i(c_i', \boldsymbol{c}_{-i}), \qquad \text{for all } c_i' \in \mathcal{C}_i.$$

Example: Consider the Matching Pennies game.

- $BR_1(P_2 \rightsquigarrow H) = H$
- $BR_1(P_2 \rightsquigarrow T) = T$
- $BR_2(P_1 \rightsquigarrow H) = T$
- $BR_2(P_1 \rightsquigarrow T) = H$



		Player 2	
		Heads	Tails
Player 1	Heads	1, -1	-1, 1
	Tails	-1, 1	1, -1

Nash Equilibrium: A Solution Concept

No player should have the motivation to unilaterally deviate from their respective strategies!

In other words, every player picks a ***best response*** to all the other players' strategies.

Definition

Given a normal (strategic) form game $\Gamma = (\mathcal{N}, \mathcal{C}, \mathcal{U})$, we call a strategy profile $(c_1, \cdots, c_N)$ a ***pure-strategy Nash equilibrium (PSNE)*** if $u_i(c_i, \boldsymbol{c}_{-i}) \geq u_i(c_i', \boldsymbol{c}_{-i})$, for all $c_i' \in \mathcal{C}_i$, for all $i \in \mathcal{N}$.

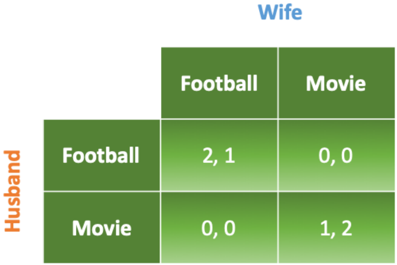
Computing PSNE: Battle of the Sexes

Description:

- Two-player coordination game.
- Husband (H): Prefers football game over movie
- Wife (W): Prefers movie over football game

Best-Response and Equilibrium Analysis:

- $BR_H(W \rightsquigarrow F) = F$
- $BR_H(W \rightsquigarrow M) = M$
- $BR_W(H \rightsquigarrow F) = F$
- $BR_W(H \rightsquigarrow M) = M$
- **PSNE:** $(F, F), (M, M)$



		Wife	
		Football	Movie
Husband	Football	2, 1	0, 0
	Movie	0, 0	1, 2

Application: Distributed Resource Allocation Games (e.g. 5G Networks)

- Tasks can be performed only when various resources (e.g. computational power, wireless spectrum) are available simultaneously.

Motivates players to form groups (or coalitions)!

Computing PSNE: Cournot's Duopoly

- Two firms produce identical item of quantities q_1 and q_2, while incurring $4c$ units of cost/quantity.
- Market clearing price: $p(q) = 100 - 2q$, where $q = q_1 + q_2$.

$$\text{Utility of the Firm-}i\text{: } u_i(q_1, q_2) = q_i \cdot p(q_1 + q_2) - 4c \cdot q_i$$

If Firm-$\{-i\}$ produces q_{-i}, then Firm-i finds its best response as follows:

$$\frac{\partial u_i(q_i, q_{-i})}{\partial q_i} = 100 - 2(q_i + q_{-i}) - 2q_i - 4c = 0.$$

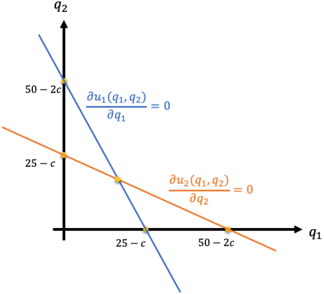
At NE, since both firms play best responses with each other, we have...

System of two best-response equations:

- $BR_1(q_2) \Rightarrow 2q_1 + q_2 = 50 - 2c$
- $BR_2(q_1) \Rightarrow q_1 + 2q_2 = 50 - 2c$

Solving them, we obtain

$$q_1^* = q_2^* = \frac{50 - 2c}{3}$$



Computing PSNE: Potential Games

Definition

A function $\Phi : \mathcal{C} \to \mathbb{R}$ is called an ***ordinal potential function*** for the game Γ, if for all $i \in \mathcal{N}$ and all $c_{-i} \in \mathcal{C}_{-i}$,

$$u_i(c, c_{-i}) - u_i(c', c_{-i}) > 0, \text{ iff } \Phi(c, c_{-i}) - \Phi(c', c_{-i}) > 0, \text{ for all } c, c' \in \mathcal{C}_i.$$

Definition

A function $\Phi : \mathcal{C} \to \mathbb{R}$ is called an ***exact potential function*** for the game Γ, if for all $i \in \mathcal{N}$ and all $c_{-i} \in \mathcal{C}_{-i}$,

$$u_i(c, c_{-i}) - u_i(c', c_{-i}) = \Phi(c, c_{-i}) - \Phi(c', c_{-i}) > 0, \text{ for all } c, c' \in \mathcal{C}_i.$$

Definition

A game Γ is called a ***potential game*** if it admits a potential function.

Theorem: [Moderer and Shapley, 1996]

Every finite ordinal potential game has a PSNE.

Example: Congestion Games

Definition

A ***congestion model*** M is defined as a tuple $(\mathcal{N}, \mathcal{R}, \mathcal{C}, \boldsymbol{x})$, where

- $\mathcal{N} = \{1, \cdots, N\}$ is the set of players
- $\mathcal{R} = \{1, \cdots, K\}$ is the set of resources
- $\mathcal{C} = \mathcal{C}_1 \times \cdots \times \mathcal{C}_N$, where $\mathcal{C}_i$ consists of sets of resources that player i can take.
- $\boldsymbol{x} = \{x_1(\ell), \cdots, x_K(\ell)\}$, where $x_k(\ell)$ is the cost of each user who uses k^{th} resource when a total of ℓ users are using it.

- Congestion games arise when users share resources to complete a given task.
  - Examples: Drivers share roads in a transportation network.

Definition

Based on the congestion model M, a ***congestion game*** is defined as $\Gamma = (\mathcal{N}, \mathcal{C}, \mathcal{U})$, with $u_i(c_i, c_{-i}) = \sum_{k \in c_i} x_k(\ell_k)$, where ℓ_k is the number of users of resource k under strategy $c = \{c_i, c_{-i}\}$.

Example: Congestion Games (cont...)

Theorem: [Rosenthal, 1973]

Every congestion game is a potential game.

Rosenthal's Potential function: For every strategy profile $c \in \mathcal{C}$, define

$$\Phi(c) = \sum_{k \in \mathcal{R}} \left(\sum_{\ell=1}^{\ell_k(c)} x_k(\ell) \right).$$

Theorem: [Moderer and Shapley, 1996]

Every potential game can be equivalently mapped to a congestion game.

Note: Usually, congestion games in transportation are modeled with large number of players ($N \to \infty$). In such a case, NE in the presence of infinitesimal players is referred to as ***Wardrop Equilibrium***.

Existence of Nash Equilibrium

Claim: PSNE may not always exist in a normal-form game!

Example: Matching Pennies

Definition

Given a normal (strategic) form game $\Gamma = (\mathcal{N}, \mathcal{C}, \mathcal{U})$, we call a mixed-strategy profile $(\pi_1, \cdots, \pi_N)$ as a ***mixed-strategy Nash equilibrium (MSNE)*** if $u_i(\pi_i, \boldsymbol{\pi}_{-i}) \geq u_i(\pi_i', \boldsymbol{\pi}_{-i})$, for all $\pi_i' \in \Delta(\mathcal{C}_i)$, for all $i \in \mathcal{N}$.

Theorem: [Nash 1951]

There always exists a MSNE in any finite normal-form game.

How to find MSNE?

Computing MSNE

Definition

Given a normal (strategic) form game $\Gamma = (\mathcal{N}, \mathcal{C}, \mathcal{U})$ and a mixed strategy π_i at the i^{th} player, the ***support*** of π_i, denoted as $\delta(\pi_i)$, is the set of all pure strategies of the i^{th} player which have non-zero probabilities, i.e.,

$$\delta(\pi_i) \triangleq \{ c \in \mathcal{C}_i \mid \pi_i(c) > 0 \}.$$

Note: Although there are uncountably infinite number of mixed strategies, there can only be finitely many supports of Nash Equilibria (NE), which is

$$\left(2^{|\mathcal{C}_1|} - 1\right) \times \cdots \times \left(2^{|\mathcal{C}_N|} - 1\right)$$

Idea: *Consider each support at a time and search for NE.*

Computing MSNE (cont...)

Theorem

The mixed strategy profile $(\pi_1, \cdots, \pi_N)$ is a NE *if and only if*, for all $i \in \mathcal{N}$,

(C1) $u_i(c, \pi_{-i})$ is the same $\forall\, c \in \delta(\pi_i)$, and

(C2) $u_i(c, \pi_{-i}) \geq u_i(c', \pi_{-i}), \ \forall\, c \in \delta(\pi_i), \ \forall\, c' \notin \delta(\pi_i)$.

If NE exists in the support $\mathcal{X}_1 \times \cdots \times \mathcal{X}_N$, where $\mathcal{X}_i = \delta(\pi_i)$, then there exists numbers $w_1, \cdots, w_N$ and mixed strategies $\pi_1, \cdots, \pi_N$ such that

$$(1) \quad w_i = \sum_{c_{-i} \in \mathcal{C}_{-i}} \left(\prod_{j \neq i} \pi_j(c_j) \right) u_i(c_i, c_{-i}), \ \forall\, c_i \in \mathcal{X}_i, \ \forall\, i \in \mathcal{N},$$

$$(2) \quad w_i \geq \sum_{c_{-i} \in \mathcal{C}_{-i}} \left(\prod_{j \neq i} \pi_j(c_j) \right) u_i(c_i, c_{-i}), \ \forall\, c_i \in \mathcal{C}_i - \mathcal{X}_i, \ \forall\, i \in \mathcal{N}.$$

$$(1) \Rightarrow \sum_{i=1}^{N} |\mathcal{X}_i| \text{ eqns, and } (2) \Rightarrow \sum_{i=1}^{N} |\mathcal{C}_i - \mathcal{X}_i| \text{ eqns.}$$

Computing MSNE...

We also need to ensure the definition of support, i.e.,

(3) $\pi_i(c) > 0, \ \forall \ c \in \mathcal{X}_i, \ \forall \ i \in \mathcal{N}$,

(4) $\pi_i(c) = 0, \ \forall \ c \in \mathcal{C}_i - \mathcal{X}_i, \ \forall \ i \in \mathcal{N}$,

(5) $\sum\limits_{c \in \mathcal{C}_i} \pi_i(c) = 1, \ \forall \ i \in \mathcal{N}$.

(3) $\Rightarrow \sum\limits_{i=1}^{N} |\mathcal{X}_i|$ eqns, (4) $\Rightarrow \sum\limits_{i=1}^{N} |\mathcal{C}_i - \mathcal{X}_i|$ eqns, and (5) $\Rightarrow N$ eqns.

Find $w_1, \cdots, w_N$ and $\pi_1, \cdots, \pi_N$ such that Equations (1)-(5) hold true.

- #(variables) $= N + \sum\limits_{i \in \mathcal{N}} |\mathcal{C}_i|$, #(equations) $= N + 2 \sum\limits_{i \in \mathcal{N}} |\mathcal{C}_i|$
- Two-Player Games ⇒ Linear Complementarity Problem (LCP)
- N-Player Games $(N > 2)$ ⇒ Non-Linear Complementarity Problem (NLCP)

Hence, computing NE in general games is HARD!

However, NE for 2-player zero-sum games can be found efficiently!

Algorithms to Compute MSNE

- Two-player zero-sum games ⇒ Linear Programming (LP)
- Two-player general-sum games ⇒ Lemke's Method
- N-player general-sum games ⇒ Lemke-Howson's Method (along many others).

This is still an active research topic!

In this course, we will only cover one algorithm for solving two-player zero-sum games.

Games & Linear Programming

This algorithm works only for two-player zero-sum games!

Before we solve games, let us build some background knowledge in linear programming!

Linear Programming (LP)

Minimize a linear function in the presence of a linear constraints.

Problem: Primal (P)

$$\begin{aligned} \underset{x \in \mathbb{R}}{\text{minimize}} \quad & c^T x \\ \text{subject to} \quad & 1.\ Ax = b, \\ & 2.\ x \succeq 0. \end{aligned}$$

Solution:

- No closed form solution
- Reliable/Efficient algorithms (Run time: $O(n^2 m)$ if $m \geq n$.)
- Software Packages: CVX

LP and Duality

Definition

The Lagrangian function is defined as

$$\begin{aligned} L(x, \lambda, \mu) &= c^T x + \lambda^T (Ax - b) - \mu^T x \\ &= -b^T \lambda + \left(A^T \lambda + c - \mu \right)^T x \end{aligned}$$

- Weighted sum of objective function and constraints.
- λ, μ: Lagrangian multipliers

Definition

The Lagrangian dual function is defined as

$$g(\lambda, \mu) = \min_{x \in \mathbb{R}} L(x, \lambda, \mu) = \begin{cases} -b^T \lambda, & \text{if } A^T \lambda + c - \mu = 0 \\ -\infty, & \text{otherwise.} \end{cases}$$

LP and Duality (cont...)

Lower Bound Property: If $\lambda \geq 0$, for any $x \in \mathbb{R}$, we have

$$c^T x \geq L(x, \lambda, \mu) \geq \min_x L(x, \lambda, \mu) = g(\lambda, \mu)$$

In other words, if v_P^* is the optimal value of the primal problem P, then, for any $\mu \succeq 0$ and $\lambda \succeq 0$, we also have $v_P^* \geq g(\lambda, \mu)$.
In other words,

$$v_P^* \geq -b^T \lambda, \text{ if } A^T \lambda + c \succeq 0.$$

Problem: Dual (D)

$$\begin{array}{ll} \underset{\lambda,\mu}{\text{maximize}} & g(\lambda, \mu) \\ \text{subject to} & 1.\ \mu \succeq 0 \\ & 2.\ \lambda \succeq 0 \end{array}$$

$$\Rightarrow$$

$$\begin{array}{ll} \underset{\lambda}{\text{maximize}} & -b^T \lambda \\ \text{subject to} & 1.\ A^T \lambda + c \succeq 0 \\ & 2.\ \lambda \succeq 0 \end{array}$$

LP and Duality (cont...)

Let v_D^* is the optimal value of the dual problem D.

Note that, $v_P^* \geq v_D^*$ always holds true.

Strong Duality: $v_P^* = v_D^*$.

- Holds true for linear programs as long as there exists a feasible point x in the search space (Slater's constraint qualifications).

Solution Methods:

- Simplex Method
- Interior-point Method
- Ellipsoid Method
- Cutting-plane Method

Python Packages for Solving LPs

- **scipy.optimize.linprog**
  - interior-point (default)
  - revised simplex
  - simplex (legacy)
- **PuLP package** (relies on CPLEX, COIN, gurobi solvers)
  - interior-point
  - revised simplex
- **CVXPY** (recommended, open source)
  - interior-point (CVXOPT/ECOS)
  - first-order optimization (SCS – parallelism with OpenMP)

Provides optimal solution to the dual problem as a certificate!

LP & Game Theory

- Let Alice's (row-player) utility matrix be U of size $m \times n$.
- Therefore, Bob's utility matrix is $-U$.
- Let Alice's and Bob's mixed strategies be a and b respectively.
- Expected utility at Alice = $a^T U b$.
- Alice's goal: $\min_b \left(\max_a a^T U b \right)$

Note: $\max_a a^T U b = \max_i e_i^T U b \triangleq \eta$,
where e_i is a vector of all zeros except for a one in the i^{th} position.
Alice's worst-case strategy can be found by solving

Problem: Alice's Primal

$$\begin{array}{ll} \underset{\eta \in \mathbb{R}, b \in \mathbb{R}^n}{\text{minimize}} & \eta \\ \text{subject to} & 1.\ \eta \mathbf{1} \succeq Ub, \text{ for all } i = 1, \cdots, \\ & 2.\ \mathbf{1}^T b = 1 \\ & 3.\ b \succeq 0. \end{array}$$

LP & Game Theory (cont...)

Define $x = \begin{bmatrix} b \\ \eta \end{bmatrix}$. Then, Alice's primal can be equivalently written as:

Problem: Alice's Primal 2

$$\begin{array}{ll} \underset{x \in \mathbb{R}^{n+1}}{\text{minimize}} & e_{n+1}^T x \\ \text{subject to} & 1. \begin{bmatrix} -U & \mathbf{1} \\ I_n & \mathbf{0} \end{bmatrix} x \succeq 0. \\ & 2. \begin{bmatrix} \mathbf{1}^T & 0 \end{bmatrix} x = 1. \end{array}$$

Lagrangian function:

$$L(x, \lambda, \mu) = e_{n+1}^T x - \lambda^T \begin{bmatrix} -U & \mathbf{1} \\ I_n & \mathbf{0} \end{bmatrix} x + \mu \left\{ \begin{bmatrix} \mathbf{1}^T & 0 \end{bmatrix} x - 1 \right\}.$$

Lagrangian dual:

$$\begin{array}{rcl} g(\lambda, \mu) & = & \min_x L(x, \lambda, \mu) \\ & = & \begin{cases} -\mu, & \text{if } e_{n+1} - \begin{bmatrix} -U^T & I_n \\ \mathbf{1}^T & \mathbf{0} \end{bmatrix} \lambda + \mu \begin{bmatrix} \mathbf{1} \\ 0 \end{bmatrix} = 0 \\ -\infty, & \text{otherwise.} \end{cases} \end{array}$$

LP & Game Theory (cont...)

Since $e_{n+1}^T x \geq L(x, \lambda, \mu) \geq g(\lambda, \mu)$, we have $e_{n+1}^T x^* \geq g(\lambda, \mu), \ \forall \ \lambda \succeq 0, \ \forall \ \mu \succeq 0.$

Problem: Alice's Dual

minimize $-\mu$

subject to 1. $\begin{bmatrix} -U^T & I_n \\ \mathbf{1}^T & \mathbf{0} \end{bmatrix} \lambda = e_{n+1} + \mu \begin{bmatrix} \mathbf{1} \\ 0 \end{bmatrix}.$

Equivalently, if we let $\hat{b} = \lambda_{-n}$ (λ without the last n entries), we have

Problem: Alice's Dual 2

minimize $-\mu$

subject to 1. $-U^T \hat{b} \succeq \mu \mathbf{1},$

2. $\mathbf{1}^T \hat{b} = 1,$

3. $\hat{b} \succeq 0.$

Claim

Alice's dual problem is equivalent to Bob's primal problem.

Summary

- *Representation:* How to represent games mathematically?
- *Information Asymmetry:* What causes information sets to exist in games?
- *Transformation:* How to represent extensive-form games in normal-form?
- *Solution Concepts:* What do we mean by solving a game?
- *Computing Equilibria:* How can we find solutions to a game?