MS&E 233
Game Theory, Data Science and AI

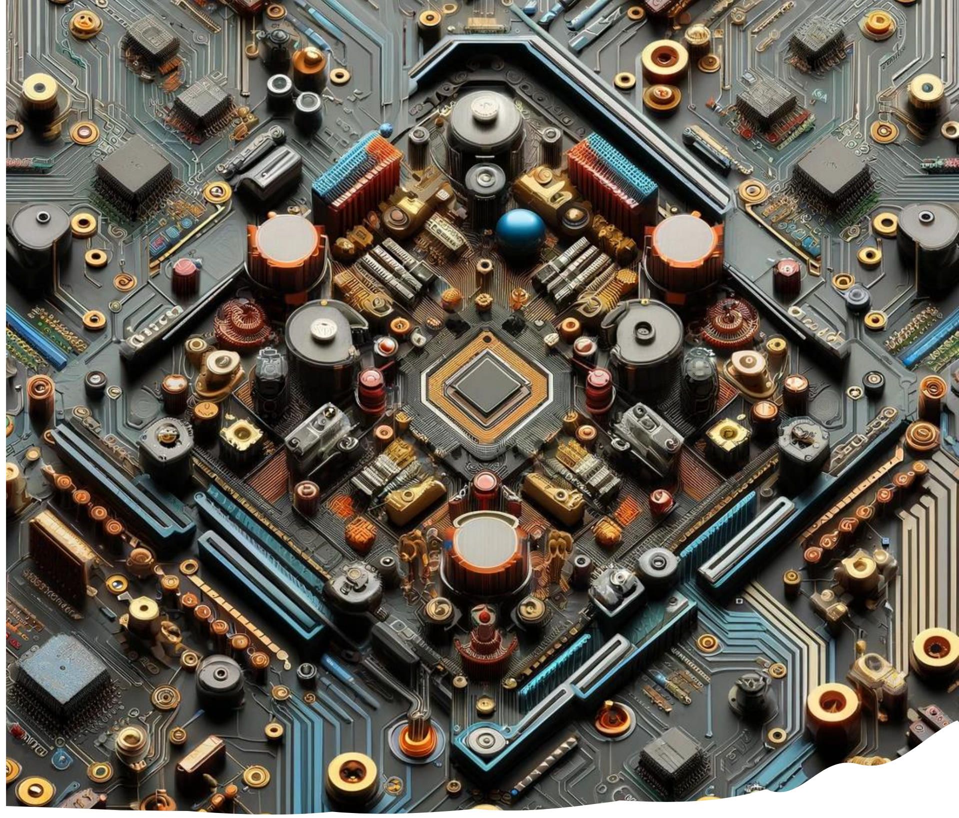
Vasilis Syrgkanis

Assistant Professor

Management Science and Engineering

(by courtesy) Computer Science and Electrical Engineering

Institute for Computational and Mathematical Engineering



Traditional engineered systems

Image credits: chat.openai.com

Engineered systems of the present

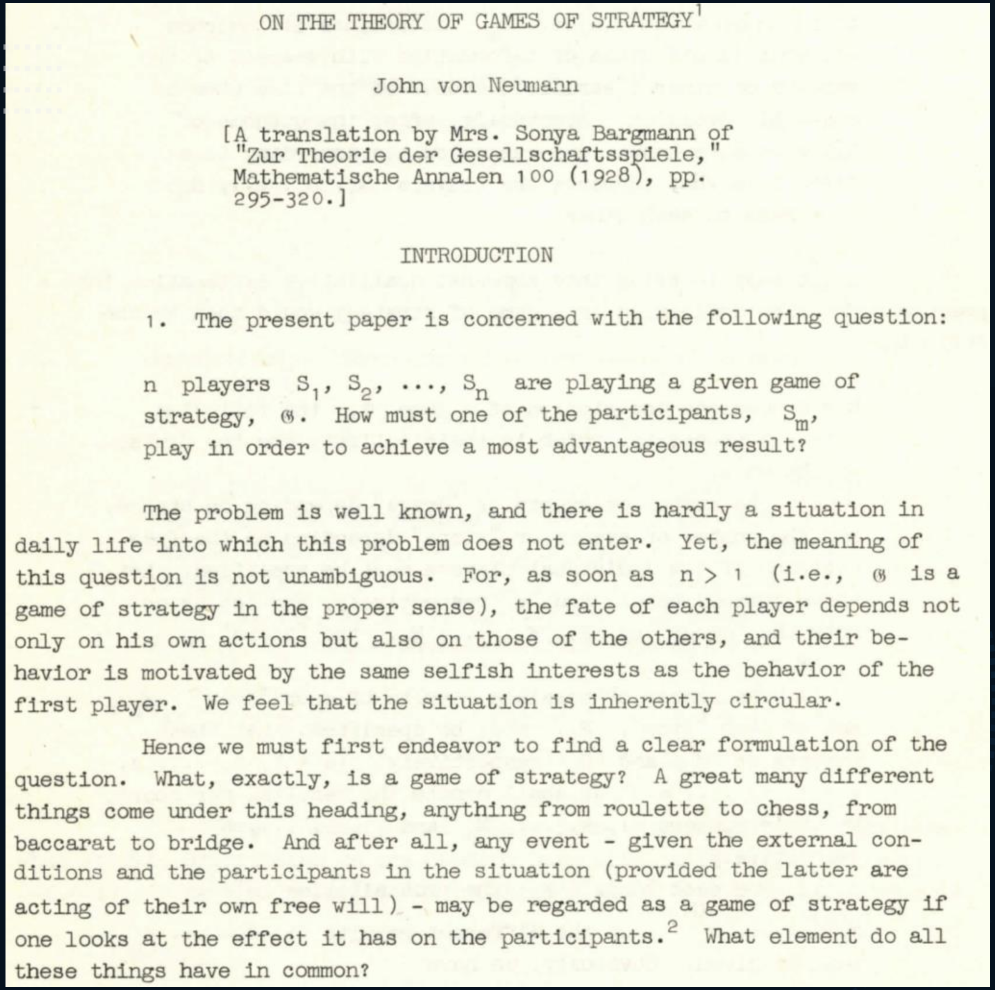
Image credits: chat.openai.com

Platforms that enable the interaction of many self-interested agents

Image credits: chat.openai.com

Game Theory

The study of mathematical models of the interaction of self-interested strategic and rational individuals



ON THE THEORY OF GAMES OF STRATEGY[1]

John von Neumann

[A translation by Mrs. Sonya Bargmann of "Zur Theorie der Gesellschaftsspiele," Mathematische Annalen 100 (1928), pp. 295-320.]

INTRODUCTION

1. The present paper is concerned with the following question:

n players $S_1, S_2, \ldots, S_n$ are playing a given game of strategy, $\mathfrak{G}$. How must one of the participants, S_m, play in order to achieve a most advantageous result?

The problem is well known, and there is hardly a situation in daily life into which this problem does not enter. Yet, the meaning of this question is not unambiguous. For, as soon as $n > 1$ (i.e., $\mathfrak{G}$ is a game of strategy in the proper sense), the fate of each player depends not only on his own actions but also on those of the others, and their behavior is motivated by the same selfish interests as the behavior of the first player. We feel that the situation is inherently circular.

Hence we must first endeavor to find a clear formulation of the question. What, exactly, is a game of strategy? A great many different things come under this heading, anything from roulette to chess, from baccarat to bridge. And after all, any event - given the external conditions and the participants in the situation (provided the latter are acting of their own free will) - may be regarded as a game of strategy if one looks at the effect it has on the participants.[2] What element do all these things have in common?

What has changed

- Real-world strategic games have become inherently complex
- Very hard for participants to strategize and fully optimize
- Many of these games take place in the digital world where automated game playing is feasible
- Large datasets that document these strategic interactions
- Advent of modern ML and AI techniques and large-scale compute

- Many opportunities and challenges at the intersection of game theory, data science and AI

Game Theory and AI

- Design of automated self-interested agents that can optimize in complex competitive environments
    - Solve complex recreational games (e.g. poker, alphaGo, multi-agent RL)
    - Strategize in complex opaque real-world games (e.g. ad auctions)

- Desiderata of an AI system framed as solutions to an equilibration process, as opposed to solutions to an optimization process
    - Many modern approaches in ML and AI can be framed as finding the solution to a complex game (e.g. boosting, generative adversarial networks, fair ML, causal ML)

Game Theory and Data Science

- How do we optimize the design of games, platforms and mechanisms for the interaction of self-interested individuals in a data driven manner?
- How do we analyze data that stem from the interaction of self-interested individuals in a competitive environment?
- How do we design and analyze experiments to measure different design choices in platforms and markets?

Course Learning Objectives

- Learn the fundamentals of game theory
- Learn how game theory can be applied in many real-world settings (e.g. ad auctions, complex games)
- Learn the fundamentals of tools from data science and ML that are useful in game theoretic contexts (online learning theory, statistical learning theory, econometrics)
- Learn how these topics can be combined to
  - provide computational solutions to the design of agents that perform well in competitive environments
  - optimize and analyze markets, mechanisms and platforms from data
- Be able to implement and code up these solutions in Python

Course Logistics

- **Grading:** 100% Homework. 7-8 HW assignments (weekly). Mostly coding in python, with some mathematical derivations.
- **HW policy:** You can collaborate in pairs, but each write their own solutions, mentioning collaborator.
- **Course webpage:** https://stanford-msande233.github.io/spring24
- **Course Discussions:** https://edstem.org/us/courses/57750/discussion/
- **Homework Submission:** https://www.gradescope.com/courses/760253
- **Compute:** HW will be released as Colab notebooks (recommended environment)
- **Gen AI:** Can use auto-completion support within Colab. You cannot use other tools.

Course Team Office Hours

- Vasilis Syrgkanis, Instructor, OF: Fri: 11-12 (Huang 252)
- Keertana Chidambaram, CA, MS&E PhD student, OF: Tue 12:45-2:45
- Gordon Martinez-Piedra, CA, EE MS student, OF: Mon 12:45-2:45

Pre-requisites

- There are no formal pre-requisites for the class
- The course will assume
    - Undergraduate level knowledge of probability, statistics, linear algebra, calculus (derivatives, integrals, partial derivatives)
    - A basic background in convex and linear optimization (first-order conditions, linear programming formalism); (*it's feasible to read along as needed; see Boyd-Vandenberghe for convex, Bertsimas-Tsitsiklis for LP*)
    - Knowledge of basic Python programming; (*we will offer however one python tutorial session at the beginning of class*)
- Familiarity with game theory concepts will be helpful but not required or assumed

Computational Game Theory for Complex Games

1
- Basics of game theory and zero-sum games (T)
- Basics of online learning theory (T)
- Solving zero-sum games via online learning (T)
- *HW1: implement simple algorithms to solve zero-sum games*
- Applications to ML and AI (T+A)
- *HW2: implement boosting as solving a zero-sum game*

2
- Basics and applications of extensive-form games (T+A)
- Solving extensive-form games via online learning (T)
- *HW3: implement agents to solve very simple variants of poker*

3
- General games and equilibria (T)
- Online learning in general games, multi-agent RL (T+A)
- *HW4: implement no-regret algorithms that converge to correlated equilibria in general games*

Data Science for Auctions and Mechanisms

4
- Basics and applications of auction theory (T+A)
- Learning to bid in auctions via online learning (T)
- *HW5: implement bandit algorithms to bid in ad auctions*

5
- Optimal auctions and mechanisms (T)
- Simple vs optimal mechanisms (T)
- *HW6: calculate equilibria in simple auctions, implement simple and optimal auctions, analyze revenue empirically*

6
- Optimizing mechanisms from samples (T)
- Online optimization of auctions and mechanisms (T)
- *HW7: implement procedures to learn approximately optimal auctions from historical samples and in an online manner*

Further Topics

7
- Econometrics in games and auctions (T+A)
- A/B testing in markets (T+A)
- *HW8: implement procedure to estimate values from bids in an auction, empirically analyze inaccuracy of A/B tests in markets*

Guest Lectures

- Mechanism Design for LLMs, Renato Paes Leme, Google Research
- Auto-bidding in Sponsored Search Auctions, Kshipra Bhawalkar, Google Research

Introduction to Game Theory

Elements of a Game

- A set of $N = \{1, \dots, n\}$ of n players, indexed typically by $i \in N$
- Each player has a space S_i of available strategies (aka actions)
- Each player chooses a strategy $s_i \in S_i$
- Given vector of strategies $s = (s_1, \dots, s_n) \in S_1 \times \cdots \times S_n$ each player receives utility $u_i(s)$, based on a utility function $u_i: S \to R$ (or receives a loss $\ell_i(s)$, based on a loss function $\ell_i: S \to R$)

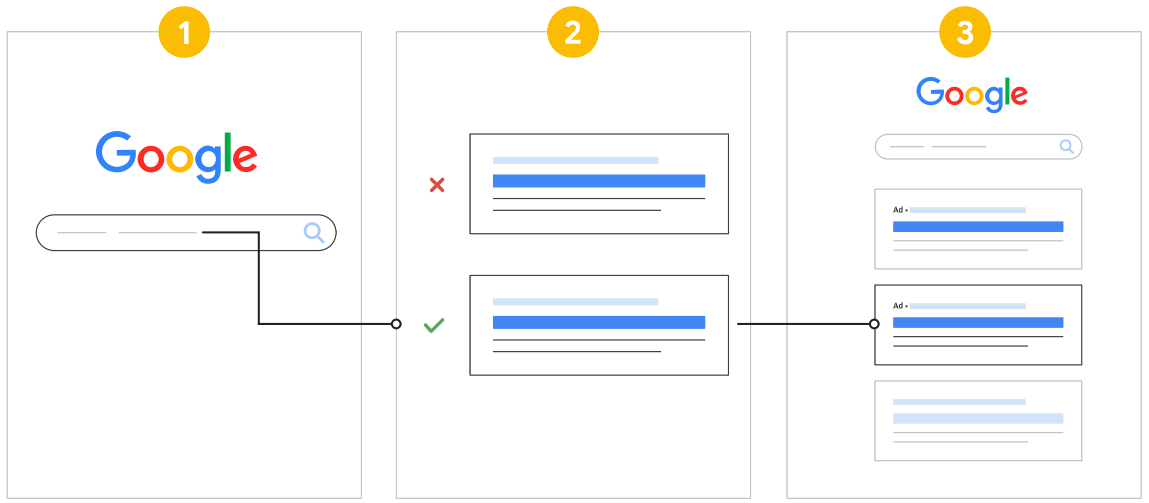
Example 1: Routing Games

- n drivers; each i wants to go from point a_i to point b_i on a road network
- Strategy space of player i: set of paths from $a_i \to b_i$
- When k_e users use road e it has latency $c_e(k_e)$
- Loss of a player: total latency on chosen path s_i

$$\ell_i(s) := \sum_{e \in s_i} c_e\big(k_e(s)\big)$$

Image credits: chat.openai.com

Example 2: Sponsored Search Auctions



- n bidders; each bidder i has an ad to display under the search for a keyword
- Strategy space of bidder i: a bid $s_i \in R$
- Bidders allocated slots in decreasing order of bids; $j_i(s)$ is slot allocated to i
- Each slot j has a probability of click x_j
- When ad gets clicked, bidder pays bid s_i
- Utility of player is net expected gains

$$u_i(s) \coloneqq x_{j_i(s)} \cdot (v_i - s_i)$$

Image credits: https://support.google.com/google-ads/answer/142918?hl=en

Example 3: Recreational Games

- Simple two-player poker
- Each players strategy is an action plan on what to do at each possible decision point in the game
- Some decisions are also being taken by "nature" randomly and only partly announced to players
- Each leaf node is an end-result and contains a utility for P1
- Utility of P1 is expected value of the terminal node that will be reached
- Utility of P2 is negative of P1 (zero-sum)

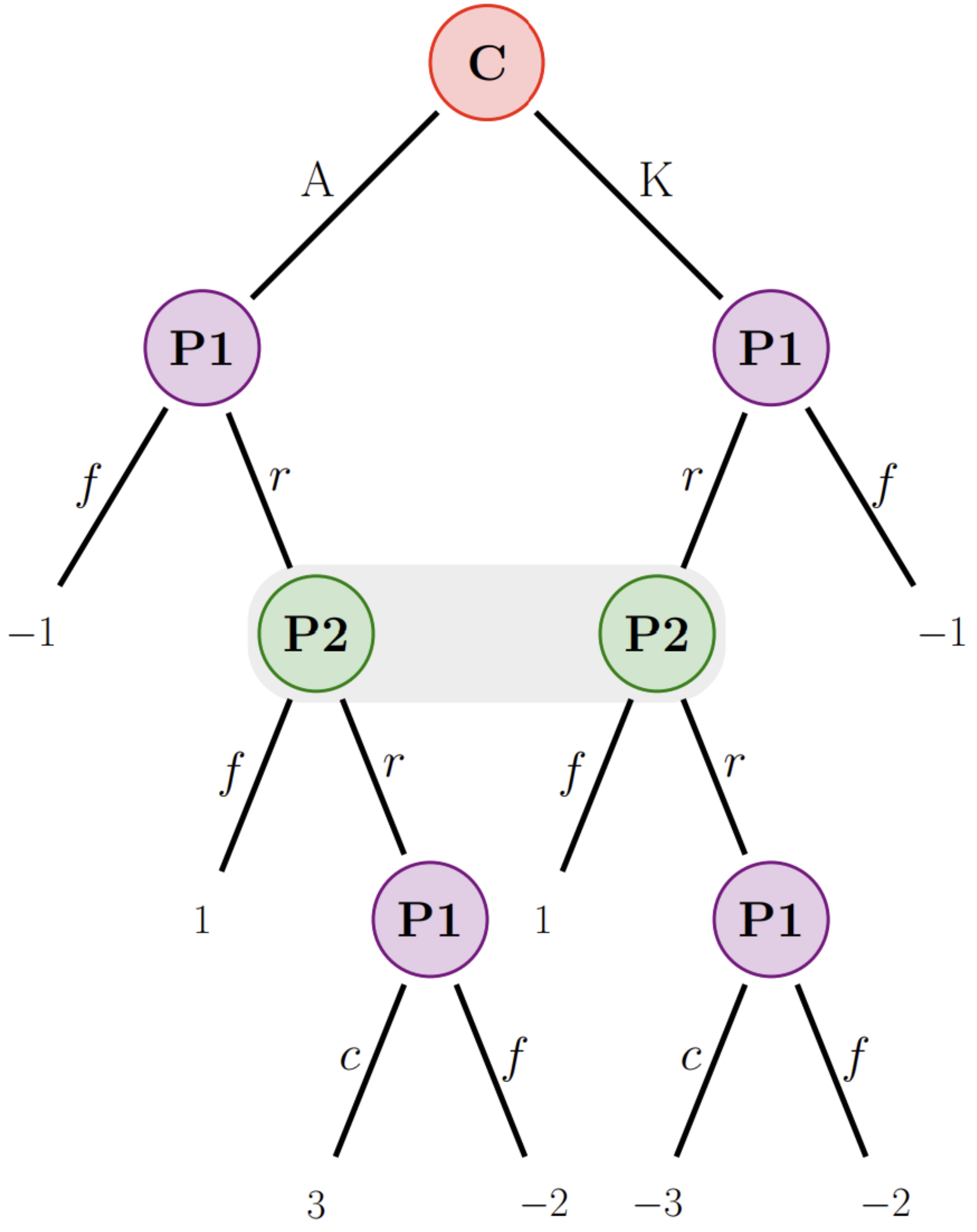


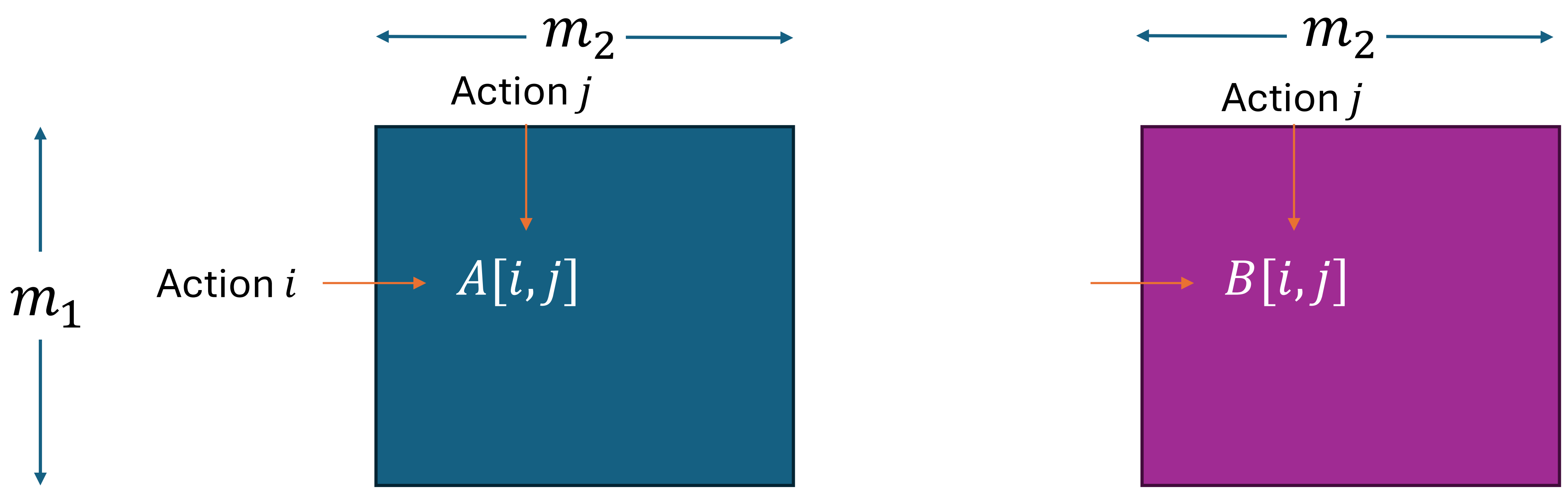
Image credits: https://www.columbia.edu/~ck2945/files/main_ai_games_markets.pdf

Finite Action Bi-Matrix Games

- Consider games with two players and finite actions
- Player one has m_1 actions and player two has m_2 actions
- We can represent the game via two $m_1 \times m_2$ matrices A, B

Finite Action Bi-Matrix Games

- Consider games with two players and finite actions
- Player one has m_1 actions and player two has m_2 actions
- We can represent the game via two $m_1 \times m_2$ matrices A, B

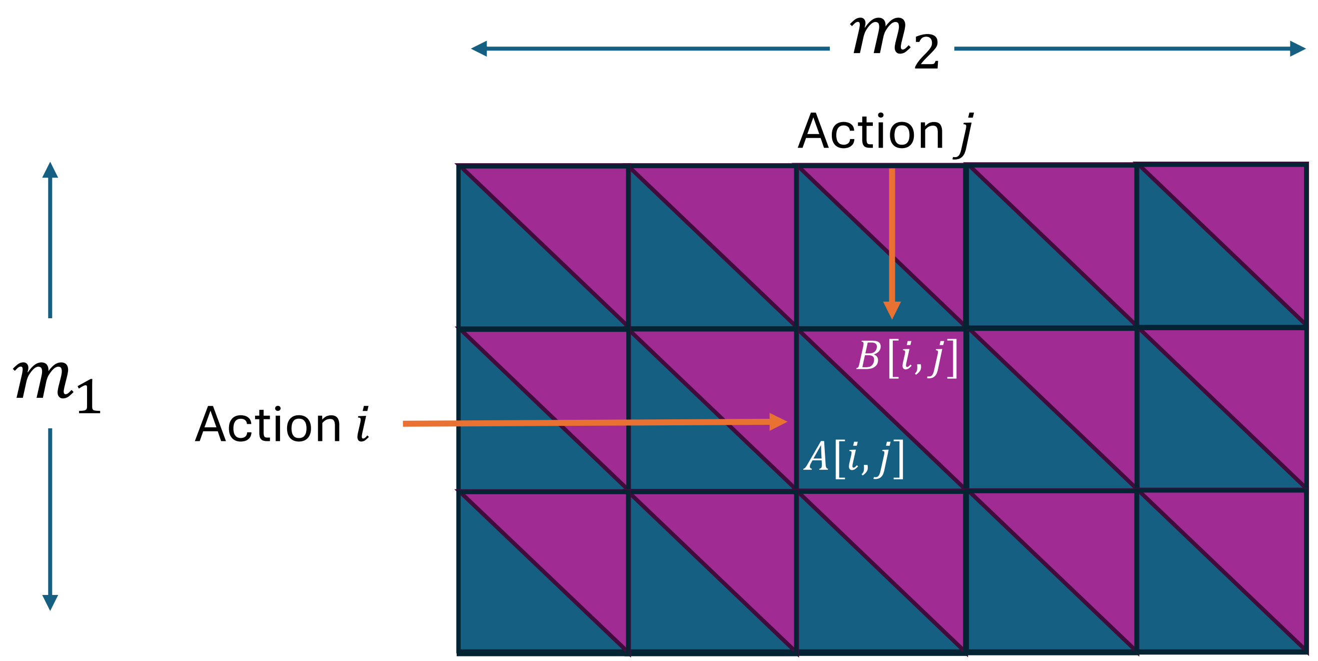


A: utility $u_1(i,j)$ of row

B: utility $u_2(i,j)$ of row

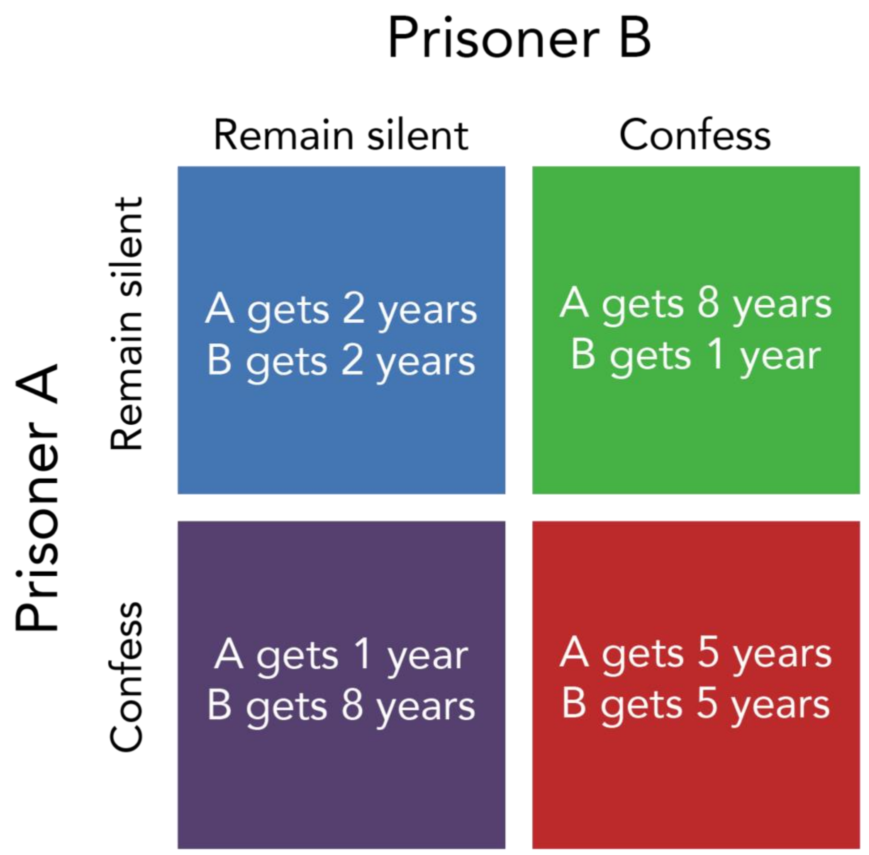
Finite Action Bi-Matrix Games

- Consider games with two players and finite actions
- Player one has m_1 actions and player two has m_2 actions
- We can represent the game via two $m_1 \times m_2$ matrices A, B



A, B joint representation

Prisoner’s Dilemma



		Prisoner B	
		Remain silent	Confess
Prisoner A	Remain silent	A gets 2 years B gets 2 years	A gets 8 years B gets 1 year
	Confess	A gets 1 year B gets 8 years	A gets 5 years B gets 5 years

How should players behave?

Dominant Strategies

Some notation

- Opponent's actions $s_{-i} = (s_1, \dots, s_{i-1}, s_{i+1}, \dots, s_n)$
- Opponent's action space: $S_{-i} = S_1 \times \cdots \times S_{i-1} \times S_{i+1} \times \cdots \times S_n$
- Utility notation: $u_i(s_i', s_{-i}) = u_i(s_1, \dots, s_{i-1}, s_i', s_{i+1}, \dots, s_n)$

Dominant Strategies

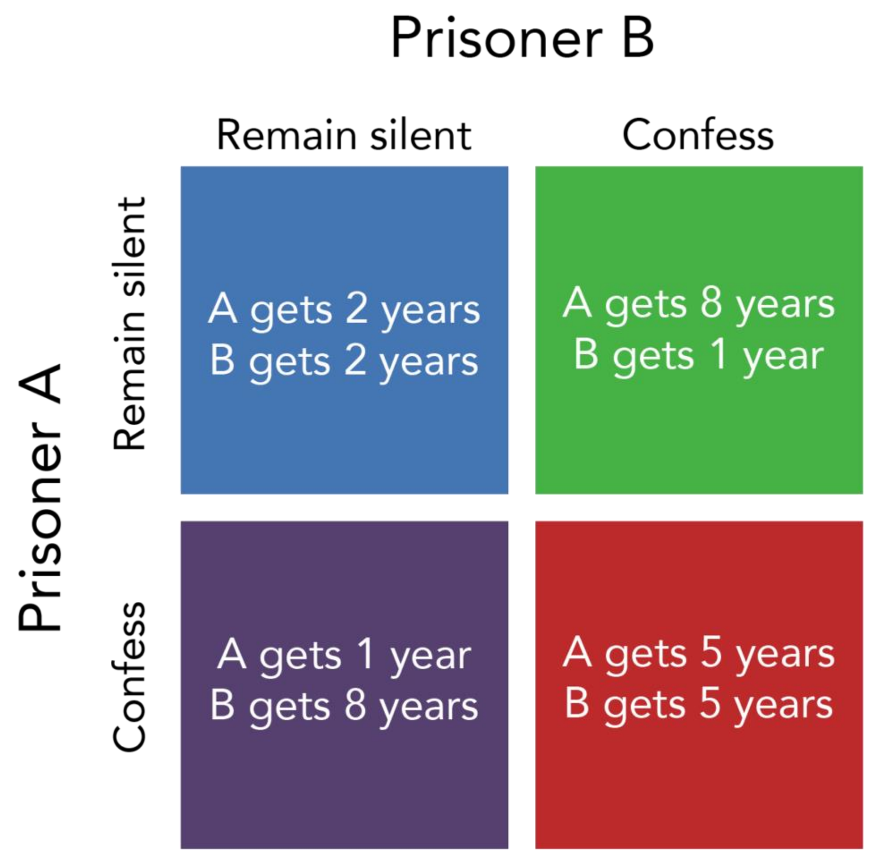
Some notation

- Opponent's actions $s_{-i} = (s_1, \dots, s_{i-1}, s_{i+1}, \dots, s_n)$
- Opponent's action space: $S_{-i} = S_1 \times \cdots \times S_{i-1} \times S_{i+1} \times \cdots \times S_n$
- Utility notation: $u_i(s_i', s_{-i}) = u_i(s_1, \dots, s_{i-1}, s_i', s_{i+1}, \dots, s_n)$

- A dominant strategy s_i is one such that no-matter what actions s_{-i} other players choose, it is always weakly better for me to choose s_i than any other strategy s_i'

$$\forall s_{-i} \in S_{-i}, \forall s_i' \in S_i: u_i(s_i, s_{-i}) \geq u_i(s_i', s_{-i})$$

Prisoner’s Dilemma



		Prisoner B	
		Remain silent	Confess
Prisoner A	Remain silent	A gets 2 years B gets 2 years	A gets 8 years B gets 1 year
	Confess	A gets 1 year B gets 8 years	A gets 5 years B gets 5 years

How should players behave?

Professor's Dilemma

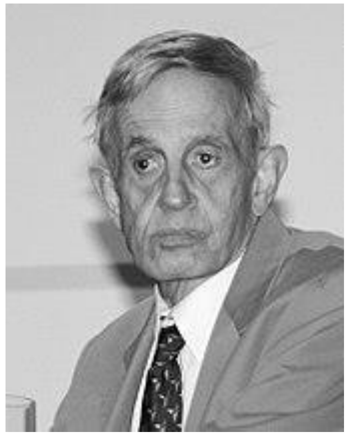
Professor \ Students	Listen	Sleep
Prepare	100, 100	-10, 0
Slack off	0, -10	0, 0

How should players behave?

Pure Nash Equilibrium

- A strategy profile $s = (s_1, \dots, s_n)$ is a pure Nash equilibrium if no player is better off, by choosing some other strategy s_i'

$$\forall s_i' \in S_i : u_i(s_i, s_{-i}) \geq u_i(s_i', s_{-i})$$

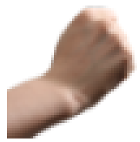
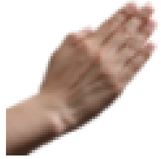
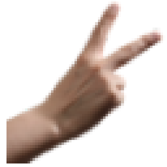
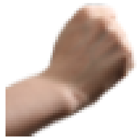
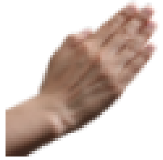
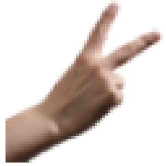


Professor’s Dilemma

		Students	
		Listen	Sleep
Professor	Prepare	100, 100	-10, 0
	Slack off	0, -10	0, 0

How should players behave?

Rock-Paper-Scissors

		Opponent		
				
		0	−1	1
You		1	0	−1
		−1	1	0

How should players behave?

Mixed Nash Equilibrium

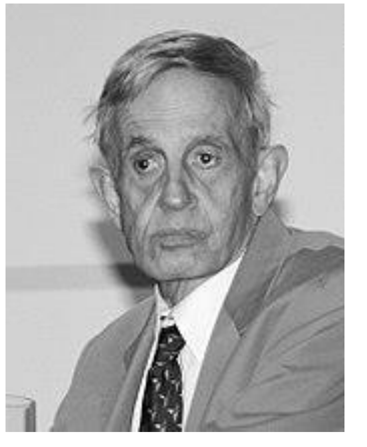
- A mixed strategy σ_i is a distribution over pure strategies
- At mixed strategy profile $\sigma = (\sigma_1, \dots, \sigma_n)$, player i gets expected utility
$$U_i(\sigma) = E_{s_1 \sim \sigma_1, \dots, s_n \sim \sigma_n}[u_i(s_1, \dots, s_n)]$$
- Utility notation: $U_i\left(s_i', \sigma_{-i}\right) = E_{s_{-i} \sim \sigma_{-i}}\left[u_i\left(s_i', s_{-i}\right)\right]$

Mixed Nash Equilibrium

- A mixed strategy σ_i is a distribution over pure strategies
- At mixed strategy profile $\sigma = (\sigma_1, \dots, \sigma_n)$, player i gets expected utility
$$U_i(\sigma) = E_{s_1 \sim \sigma_1, \dots, s_n \sim \sigma_n}[u_i(s_1, \dots, s_n)]$$
- Utility notation: $U_i(s_i', \sigma_{-i}) = E_{s_{-i} \sim \sigma_{-i}}[u_i(s_i', s_{-i})]$

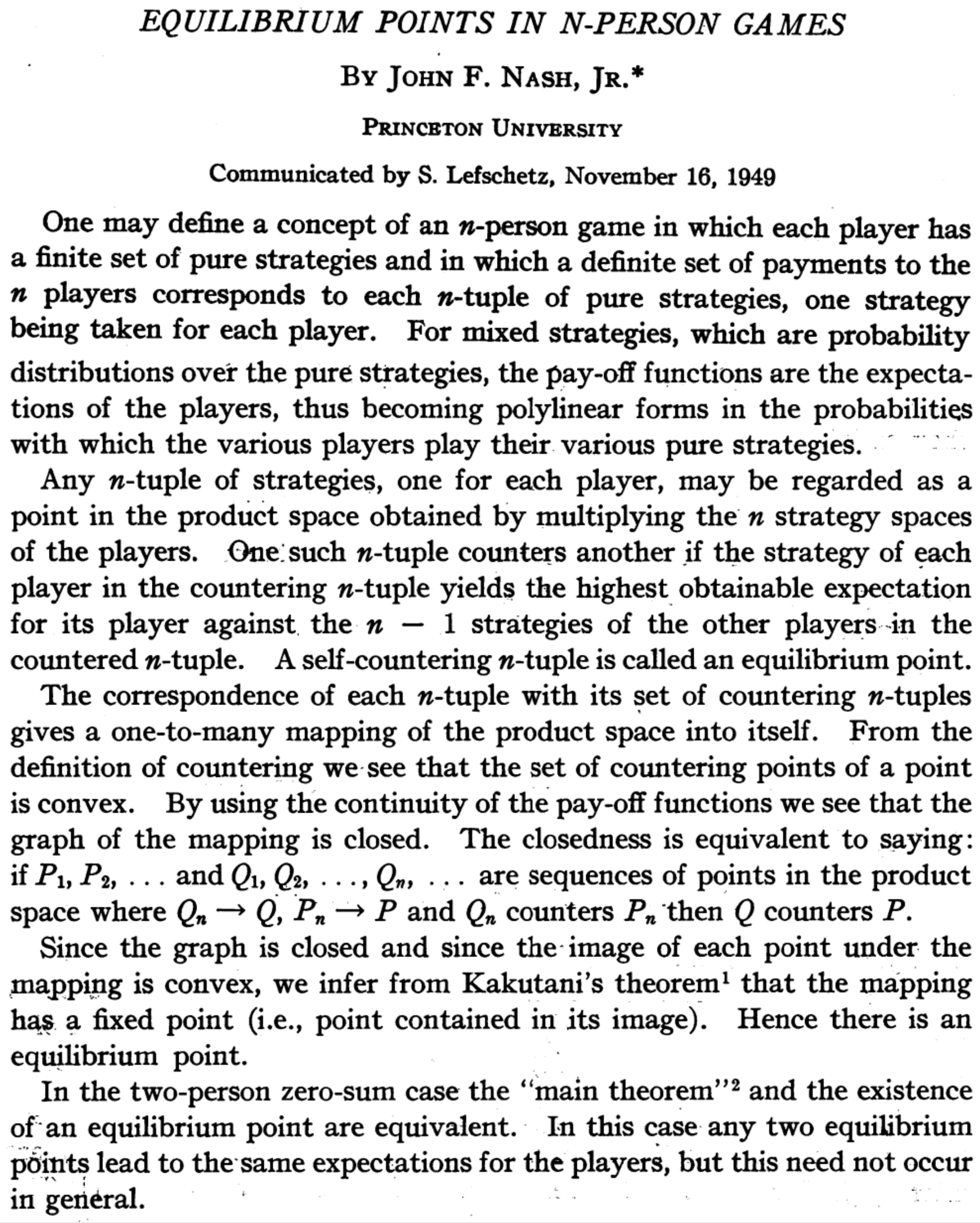
- A mixed strategy profile $\sigma = (\sigma_1, \dots, \sigma_n)$ is a Nash equilibrium if no player is better off in expectation, by choosing another strategy s_i'

$$\forall s_i' \in S_i \colon U_i(\sigma) \geq U_i(s_i', \sigma_{-i})$$



Existence of Nash Equilibrium [Nash1950]

Every n player finite action game has at least one mixed Nash equilibrium



EQUILIBRIUM POINTS IN N-PERSON GAMES

BY JOHN F. NASH, JR.*

PRINCETON UNIVERSITY

Communicated by S. Lefschetz, November 16, 1949

One may define a concept of an n-person game in which each player has a finite set of pure strategies and in which a definite set of payments to the n players corresponds to each n-tuple of pure strategies, one strategy being taken for each player. For mixed strategies, which are probability distributions over the pure strategies, the pay-off functions are the expectations of the players, thus becoming polylinear forms in the probabilities with which the various players play their various pure strategies.

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 $Q_n \rightarrow Q$, $P_n \rightarrow P$ 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.

In the two-person zero-sum case the "main theorem"[2] and the existence of an equilibrium point are equivalent. In this case any two equilibrium points lead to the same expectations for the players, but this need not occur in general.

Computation of Mixed Nash Equilibrium

- Suppose we knew the supports (C_1, C_2) of a mixed Nash equilibrium (i.e. strategies that receive positive probability)

Computation of Mixed Nash Equilibrium

- Suppose we knew the supports (C_1, C_2) of a mixed Nash equilibrium (i.e. strategies that receive positive probability)
- All actions in C_1 must yield the same expected utility for player one

$$\forall i \in C_1\colon \sum_{j \in C_2} A[i,j]\, y_j = V_1, \qquad \sum_{j \in C_2} y_j = 1$$

Computation of Mixed Nash Equilibrium

- Suppose we knew the supports (C_1, C_2) of a mixed Nash equilibrium (i.e. strategies that receive positive probability)
- All actions in C_1 must yield the same expected utility for player one

$$\forall i \in C_1: \sum_{j \in C_2} A[i,j]\, y_j = V_1, \qquad \sum_{j \in C_2} y_j = 1$$

- All actions in C_2 must yield the same expected utility for player two

$$\forall j \in C_2: \sum_{i \in C_1} x_i\, B[i,j] = V_2, \qquad \sum_{i \in C_1} x_i = 1$$

Computation of Mixed Nash Equilibrium

- Suppose we knew the supports (C_1, C_2) of a mixed Nash equilibrium (i.e. strategies that receive positive probability)
- All actions in C_1 must yield the same expected utility for player one

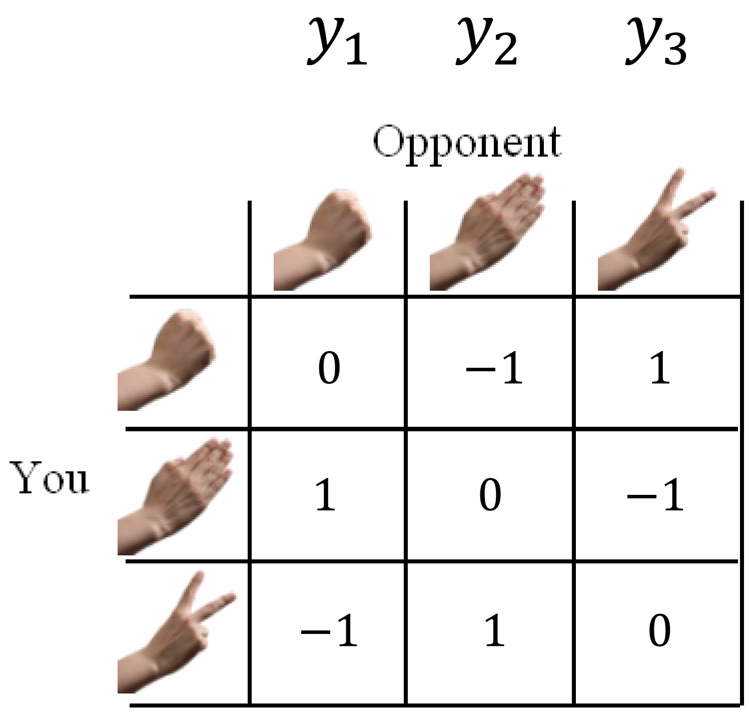
$$\forall i \in C_1 \colon \sum_{j \in C_2} A[i,j]\, y_j = V_1, \qquad \sum_{j \in C_2} y_j = 1$$

- All actions in C_2 must yield the same expected utility for player two

$$\forall j \in C_2 \colon \sum_{i \in C_1} x_i\, B[i,j] = V_2, \qquad \sum_{i \in C_1} x_i = 1$$

- System of $|C_1| + |C_2|$ linear equations with $|C_1| + |C_2|$ unknowns

Computation of Mixed Nash Equilibrium


Opponent
You

You	y_1	y_2	y_3
	0	-1	1
	1	0	-1
	-1	1	0

- Suppose we look for a full support NE
- Strategy y of column player must be such that all rows give the same utility for row player

$$-y_2 + y_3 = y_1 - y_3 = -y_1 + y_2$$

- and must be a valid distribution

$$y_1 + y_2 + y_3 = 1$$

- Combining appropriately

$$y_1 + y_2 = 2y_3 \Rightarrow 3y_3 = 1 \Rightarrow y_3 = 1/3$$
$$y_2 + y_3 = 2y_1 \Rightarrow 3y_1 = 1 \Rightarrow y_1 = 1/3$$

Intractability of Mixed Nash Equilibrium

- The assumption of knowing the supports was crucial
- For games with many actions, we cannot enumerate all possible supports (combinatorial explosion)
- Turns out there is no easy way to side-step this

- Computing a mixed NE in two player games is “intractable”

- It is provable as hard as computing a “fixed point” ($f(x) = x$) of an arbitrary function f, which is considered an intractable problem

Two Player Zero-Sum Games

- Player one (“min” player or “row” player)
- Player two (“max” player or “column” player)
- Player one has n possible actions
- Player two has m possible actions

- If player one chooses action i and player two chooses action j then player one incurs loss $A[i, j]$ and player two gains utility $A[i, j]$

Equilibrium via Min-Max Theorem

- Suppose that both players behave *pessimistically*
- Row (min) player thinks: "I'll choose a strategy x such that I'll try to minimize the worst-case loss that the other player can cause me"

$$\bar{x} = \underset{x}{\operatorname{argmin}} \left(\max_{y} x' A y \right)$$

- Column (max) player thinks: "I'll choose a strategy y such that I'll try to maximize the worst-case utility that the other player will allow me to get"

$$\bar{y} = \underset{y}{\operatorname{argmax}} \left(\min_{x} x' A y \right)$$

Equilibrium via Min-Max Theorem

- Suppose both players behave pessimistically

$$\bar{x} = \underset{x}{\operatorname{argmin}} \left(\max_{y} x' A y \right), \qquad \bar{y} = \underset{y}{\operatorname{argmax}} \left(\min_{x} x' A y \right)$$

- Can $\bar{x}, \bar{y}$ be equilibrium despite both players being pessimistic?

Equilibrium via Min-Max Theorem

- Suppose both players behave pessimistically

$$\bar{x} = \underset{x}{\operatorname{argmin}} \left(\max_{y} x' A y \right), \qquad \bar{y} = \underset{y}{\operatorname{argmax}} \left(\min_{x} x' A y \right)$$

- Can $\bar{x}, \bar{y}$ be equilibrium despite both players being pessimistic?
- What if pessimistic value that each player achieves is the same?

$$\min_{x} \max_{y} x' A y = \max_{y} \min_{x} x' A y$$

Equilibrium via Min-Max Theorem

- Suppose both players behave pessimistically

$$\bar{x} = \operatorname*{argmin}_x \left(\max_y x'Ay\right), \qquad \bar{y} = \operatorname*{argmax}_y \left(\min_x x'Ay\right)$$

- Can $\bar{x}, \bar{y}$ be equilibrium despite both players being pessimistic?
- What if pessimistic value that each player achieves is the same?

$$\min_x \max_y x'Ay = \max_y \min_x x'Ay$$

Smallest loss that min player can achieve if max chooses $\bar{y}$

$$\boxed{\bar{x}'A\bar{y}} \le \boxed{\max_y \bar{x}'Ay} = \boxed{\min_x \max_y x'Ay} = \boxed{\max_y \min_x x'Ay} = \boxed{\min_x x'A\bar{y}}$$

Loss of min player at $(\bar{x}, \bar{y})$

Pessimistic loss if I choose $\bar{x}$

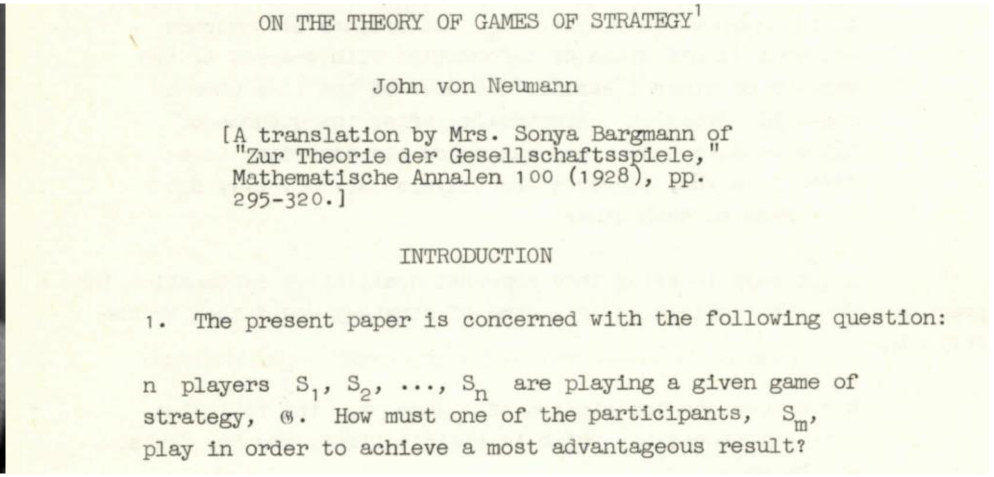
Best pessimistic loss by definition of $\bar{x}$

Best pessimistic utility that max player can achieve

Pessimistic utility that max player achieves by using $\bar{y}$

Von-Neuman’s Min-Max Theorem [1928]

$$\min_{x}\max_{y} x'Ay = \max_{y}\min_{x} x'Ay$$



ON THE THEORY OF GAMES OF STRATEGY[1]

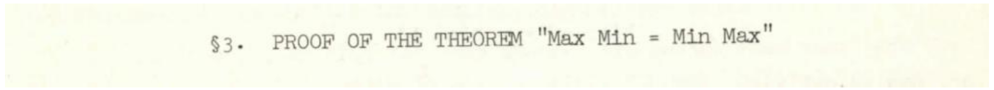
John von Neumann

[A translation by Mrs. Sonya Bargmann of "Zur Theorie der Gesellschaftsspiele," Mathematische Annalen 100 (1928), pp. 295-320.]

INTRODUCTION

1. The present paper is concerned with the following question:

n players $S_1, S_2, \ldots, S_n$ are playing a given game of strategy, $\mathfrak{G}$. How must one of the participants, S_m, play in order to achieve a most advantageous result?



§3. PROOF OF THE THEOREM "Max Min = Min Max"

Min-Max Theorem via LP-duality

- Can think of best pessimistic loss/utility as Linear Programs (LPs)

$$\begin{aligned} \min_{z\in R,\, x}\ & z \\ A'x &\le \mathbf{1}z \\ \mathbf{1}'x &= 1 \\ x &\ge 0 \end{aligned}$$

$$\begin{aligned} \max_{v\in R,\, y}\ & v \\ Ay &\ge \mathbf{1}v \\ \mathbf{1}'y &= 1 \\ y &\ge 0 \end{aligned}$$

Min-Max Theorem via LP-duality

- Can think of best pessimistic loss/utility as Linear Programs (LPs)
- Turns out, one program is the dual of the other

Find x with the smallest such loss guarantee z

$$\min_{z \in R,\, x} \; z$$

No matter what action opponent takes, loss is at most z

$$\boxed{A'x \leq \mathbf{1}z}$$
$$\mathbf{1}'x = 1$$
$$x \geq 0$$

$$\max_{v \in R,\, y} \; v$$
$$Ay \geq \mathbf{1}v$$
$$\mathbf{1}'y = 1$$
$$y \geq 0$$

Min-Max Theorem via LP-duality

- Can think of best pessimistic loss/utility as Linear Programs (LPs)
- Turns out, one program is the dual of the other
- These can be transformed into the canonical form of two dual LPs

$$\boxed{\min_{u\geq 0: Au\geq c} \; b'u}$$

$$\overbrace{\begin{gathered}\min_{z\in R,\, x} \; z \\ A'x \leq \mathbf{1}z \\ \mathbf{1}'x = 1 \\ x \geq 0\end{gathered}}$$

$$\boxed{\max_{w\geq 0: A'w\leq b} \; c'w}$$

$$\overbrace{\begin{gathered}\max_{v\in R,\, y} \; v \\ Ay \geq \mathbf{1}v \\ \mathbf{1}'y = 1 \\ y \geq 0\end{gathered}}$$

Min-Max Theorem via LP-duality

- Can think of best pessimistic loss/utility as Linear Programs (LPs)
- Turns out, one program is the dual of the other
- These can be transformed into the canonical form of two dual LPs
- By strong LP duality the two linear programs have the same value

$$\begin{array}{c} \min\limits_{z \in R,\, x} z \\ A'x \leq \mathbf{1}z \\ \mathbf{1}'x = 1 \\ x \geq 0 \end{array} \quad = \quad \begin{array}{c} \max\limits_{v \in R,\, y} v \\ Ay \geq \mathbf{1}v \\ \mathbf{1}'y = 1 \\ y \geq 0 \end{array}$$

Min-Max Theorem via LP-duality

- Can think of best pessimistic loss/utility as Linear Programs (LPs)
- Turns out, one program is the dual of the other
- These can be transformed into the canonical form of two dual LPs
- By strong LP duality the two linear programs have the same value

$$\min_x \max_y x'Ay = \max_y \min_x x'Ay$$

Appendix I

Transforming dual programs for zero-sum games into canonical forms

Min-Max Theorem via LP-duality

- Can think of best pessimistic loss/utility as Linear Programs (LPs)
- Turns out, one program is the dual of the other

$$\min_{z_1, z_2 \in R,\ x \in R^n} z_1 - z_2$$
$$A'x \leq \mathbf{1}(z_1 - z_2)$$
$$\mathbf{1}'x = 1$$
$$x, z_1, z_2 \geq 0$$

$$\max_{v_1, v_2 \in R,\ y \in R^m} v_1 - v_2$$
$$Ay \geq \mathbf{1}(v_1 - v_2)$$
$$\mathbf{1}'y = 1$$
$$y, v_1, v_2 \geq 0$$

Min-Max Theorem via LP-duality

- Can think of best pessimistic loss/utility as Linear Programs (LPs)
- Turns out, one program is the dual of the other

$$\begin{aligned} \min_{z_1, z_2 \in R,\ x \in R^n} & (0, \ldots, 0, 1, -1)'(x, z_1, z_2) \\ -A'x + \mathbf{1} z_1 - \mathbf{1}\, z_2 &\geq 0 \\ \mathbf{1}'x &\geq 1 \\ -\mathbf{1}'x &\geq -1 \\ x, z_1, z_2 &\geq 0 \end{aligned}$$

$$\begin{aligned} \max_{v_1, v_2 \in R,\ y \in R^m} & (0, \ldots, 0, 1, -1)'(y, v_1, v_2) \\ -Ay + \mathbf{1} v_1 - \mathbf{1} v_2 &\leq 0 \\ \mathbf{1}'y &\leq 1 \\ -\mathbf{1}'y &\leq -1 \\ y, v_1, v_2 &\geq 0 \end{aligned}$$

Min-Max Theorem via LP-duality

- Can think of best pessimistic loss/utility as Linear Programs (LPs)
- Turns out, one program is the dual of the other

$$\min_{z_1, z_2 \in R,\, x \in R^n} (0, \dots, 0, 1, -1)'(x, z_1, z_2) \qquad \max_{v_1, v_2 \in R,\, y \in R^m} (0, \dots, 0, 1, -1)'(y, v_1, v_2)$$

$$\begin{pmatrix} -A' & \mathbf{1} & -\mathbf{1} \\ \mathbf{1}' & 0 & 0 \\ -\mathbf{1}' & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} x \\ z_1 \\ z_2 \end{pmatrix} \geq \begin{pmatrix} 0 \\ 1 \\ -1 \end{pmatrix} \qquad \begin{pmatrix} -A & \mathbf{1} & -\mathbf{1} \\ \mathbf{1}' & 0 & 0 \\ -\mathbf{1}' & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} y \\ v_1 \\ v_2 \end{pmatrix} \geq \begin{pmatrix} 0 \\ 1 \\ -1 \end{pmatrix}$$

$$x, z_1, z_2 \geq 0 \qquad y, v_1, v_2 \geq 0$$