

L01 Introduction

CS 280 Algorithmic Game Theory

Ioannis Panageas

Course material

We will use Edstem and material will also be posted on <https://panageas.github.io/agt2026.html>

Recommended Textbooks

- Nisan/Roughgarden/Tardos/Vazirani (eds), Algorithmic Game Theory (online).
- Tim Roughgarden notes (online).

Many lectures will not be part of the above!

Grading

- **Homework**: 40%
 - There will be given 2 Homeworks to solve (**Latex!**).
- **Scribing lecture notes**: 25%
 - **Latex template**, Group of 1-2. Deadline 2 weeks after the lecture.
- **Research Project/Implementation/Present paper** : 35%
 - Group of ~3-4. Report Deadline on 7th of December via Gradescope.
 - Presentation last week of classes via zoom.

What is Game Theory?

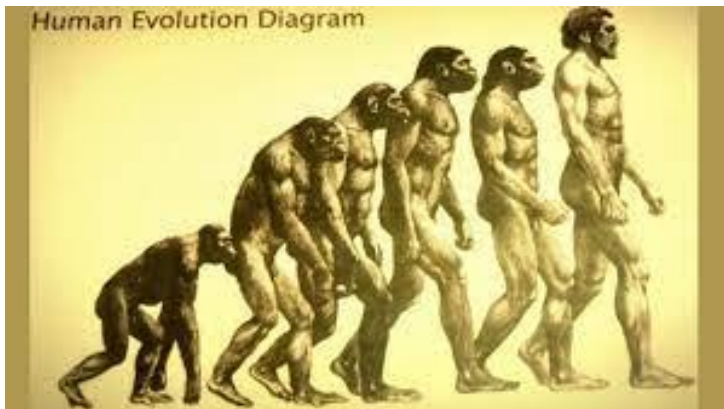
Markets - Auctions



Routing



Evolution



Elections



What is Game Theory?

Games are thought experiments helping us to *predict rational behavior* in *situations of conflict*.

1. *Conflict*: Everybody's actions affect others.
2. *Rational Behavior*: The players want to maximize their own expected utility.
3. *Predict*: We want to know what happens. Via solution concepts.

What is Game Theory?

Games are thought experiments helping us to *predict rational behavior* in *situations of conflict*.

1. *Conflict*: Everybody's actions affect others.
2. *Rational Behavior*: The players want to maximize their own expected utility.
3. *Predict*: We want to know what happens. Via solution concepts.

What is Game Theory?

Games are thought experiments helping us to *predict rational behavior* in *situations of conflict*.

1. *Conflict*: Everybody's actions affect others.
2. *Rational Behavior*: The players want to maximize their own expected utility.
3. *Predict*: We want to know what happens. Via solution concepts.

Example: Prisoner's Dilemma

Simultaneously, the police offer each prisoner a bargain:

- If A and B both confess, each of them serves 2 years in prison.
- If A confesses but B denies, A will be set free and B will serve 3 years in prison (and vice versa).
- If both A and B deny the crime, they will both serve 1 year in prison.

	Deny	Confess
Deny	1, 1	3, 0
Confess	0, 3	2, 2

Example: Prisoner's Dilemma

Simultaneously, the police offer each prisoner a bargain:

- If A and B both confess, each of them serves 2 years in prison.
- If A confesses but B denies, A will be set free and B will serve 3 years in prison (and vice versa).
- If both A and B deny the crime, they will both serve 1 year in prison.

	Deny	Confess
Deny	1, 1	3, 0
Confess	0, 3	2, 2

How A and B will play?

Example: Rock-Paper-Scissors

			
	0, 0	-1, 1	1, -1
	1, -1	0, 0	-1, 1
	-1, 1	1, -1	0, 0

Example: Rock-Paper-Scissors

			
	0, 0	-1, 1	1, -1
	1, -1	0, 0	-1, 1
	-1, 1	1, -1	0, 0

No dominant strategy equilibrium!

Concept: **Nash Equilibrium**

A pair of strategies (deterministic or randomized) such that the strategy of the row player is **at least as good** as any other strategy of her given the strategy of the column player (and vice versa).

Bimatrix Games

- 2 players: **Row** and **Column**
- n, m **strategies** available
- **Payoff** matrices R, C of size $n \times m$.

$$\mathcal{G} = (R_{n \times m}, C_{n \times m})$$



payoff of the column player
for playing j
when row player plays i .

payoff of the row player for playing i
when column player plays j .

Bimatrix Games

Column player
chooses $y \in \Delta_m$

Row player
chooses $x \in \Delta_n$


$$R_{ij}, C_{ij}$$



Row gets $x^\top R y$.
Column gets $x^\top C y$.

Bimatrix Games

Column player
chooses $y \in \Delta_m$

Row player
chooses $x \in \Delta_n$


$$R_{ij}, C_{ij}$$



Row gets $x^\top Ry$.
Column gets $x^\top Cy$.

Definition (Nash Equilibrium). (x^*, y^*) is a Nash Equilibrium iff for all possible randomized strategies x' of row player holds

$$x^{*\top} Ry^* \geq x'^\top Ry^*$$

and for all possible randomized strategies y' of column player holds

$$x^{*\top} Cy^* \geq x^{*\top} Cy'.$$

Example: Rock-Paper-Scissors

			
	0, 0	-1, 1	1, -1
	1, -1	0, 0	-1, 1
	-1, 1	1, -1	0, 0

The unique Nash Equilibrium is $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$.

Remark:

Contrary to Prisoner's Dilemma, in RPS **randomization** is **necessary** for Nash equilibrium to **exist**!

Example: Rock-Paper-Scissors

			
	0, 0	-1, 1	1, -1
	1, -1	0, 0	-1, 1
	-1, 1	1, -1	0, 0

Nice solution concept but does it always exist?

Remark:

Contrary to Prisoner's Dilemma, in RPS *randomization* is *necessary* for Nash equilibrium to *exist*!

Example: Rock-Paper-Scissors

			
	0, 0	-1, 1	1, -1
	1, -1	0, 0	-1, 1
	-1, 1	1, -1	0, 0

von Neumann '28:

*For **two-player zero-sum games**, i.e., $R + C = 0$, it
always exists!*

Remark:

Contrary to Prisoner's Dilemma, in RPS **randomization** is **necessary** for Nash equilibrium to **exist!**

Example: Modified Rock-Paper-Scissors

			
	0, 0	-1, 1	2, -1
	1, -1	0, 0	-1, 1
	-2, 1	1, -1	0, 0

Example: Modified Rock-Paper-Scissors

			
	0, 0	-1, 1	2, -1
	1, -1	0, 0	-1, 1
	-2, 1	1, -1	0, 0

Not zero sum anymore!

Example: Modified Rock-Paper-Scissors

	25% 	50% 	25% 
33.3% 	0, 0	-1, 1	2, -1
33.3% 	1, -1	0, 0	-1, 1
33.3% 	-2, 1	1, -1	0, 0

Not zero sum anymore!

John Nash '51:

There always **exists** a **Nash equilibrium** (finite games)!

Cool but **Algorithmic**?



Question: Can we predict what will happen in a **large** system?

- ***Computing Nash Equilibrium:*** Design fast Algorithms to compute Nash Equilibrium!
- ***Mechanism Design:*** Design a system that will be used by users to optimize our objectives!

Cool but Algorithmic?



Question: Can we predict what will happen in a **large** system?

Game theory: Yes, via solution concept (system will reach equilibrium).

- ***Computing Nash Equilibrium:***
Design fast Algorithms to compute Nash Equilibrium!
- ***Mechanism Design:*** Design a system that will be used by users to optimize our objectives!

Cool but Algorithmic?



Question: Can we predict what will happen in a **large** system?

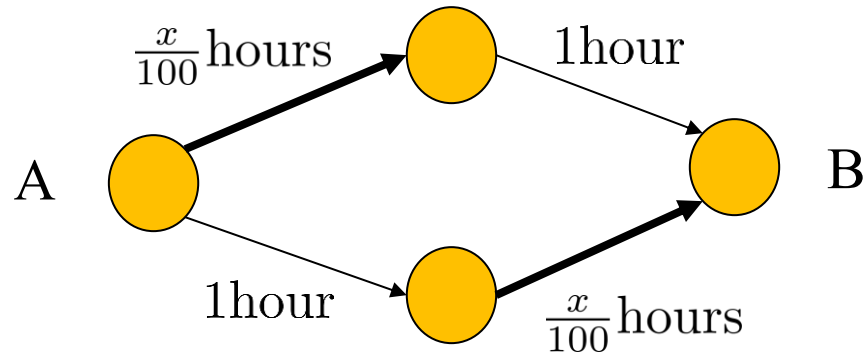
Game theory: Yes, via solution concept (system will reach equilibrium).

Question: How to **compute efficiently** an equilibrium?

- ***Computing Nash Equilibrium:***
Design fast Algorithms to compute Nash Equilibrium!
- ***Mechanism Design:*** Design a system that will be used by users to optimize our objectives!

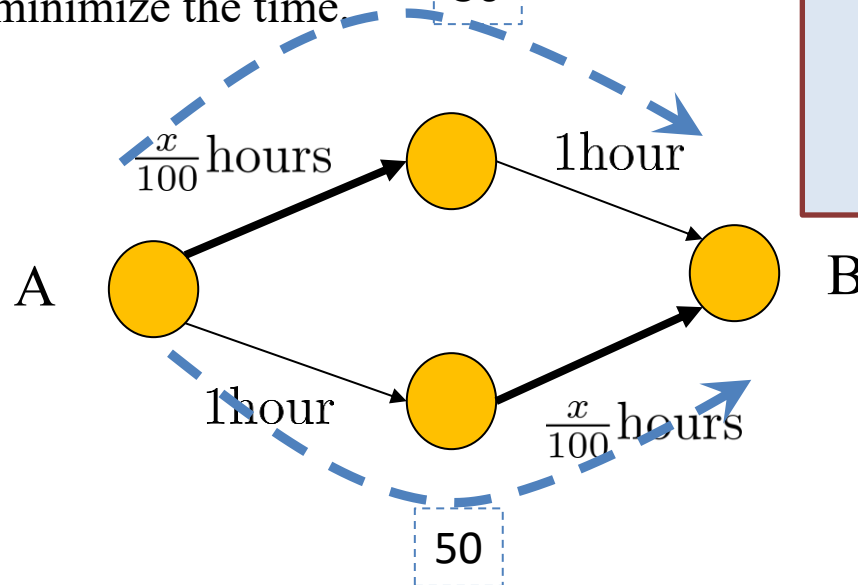
Price of Anarchy

Suppose 100 drivers commute from A to B.
Drivers want to minimize the time.



Price of Anarchy

Suppose 100 drivers commute from A to B.
Drivers want to minimize the time.

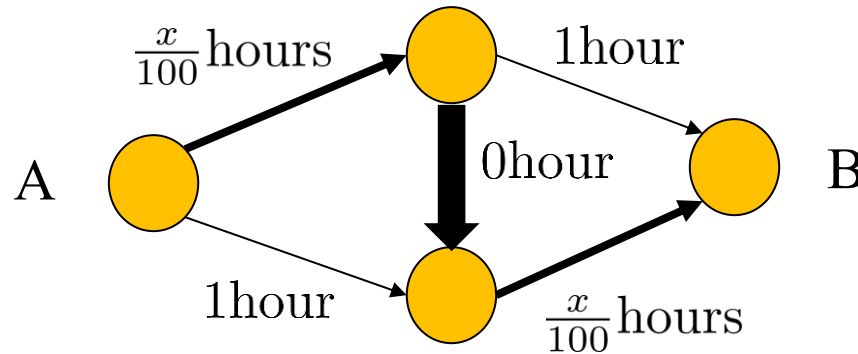


Delay is 1.5 hours for everybody at the unique Nash equilibrium.

Price of Anarchy

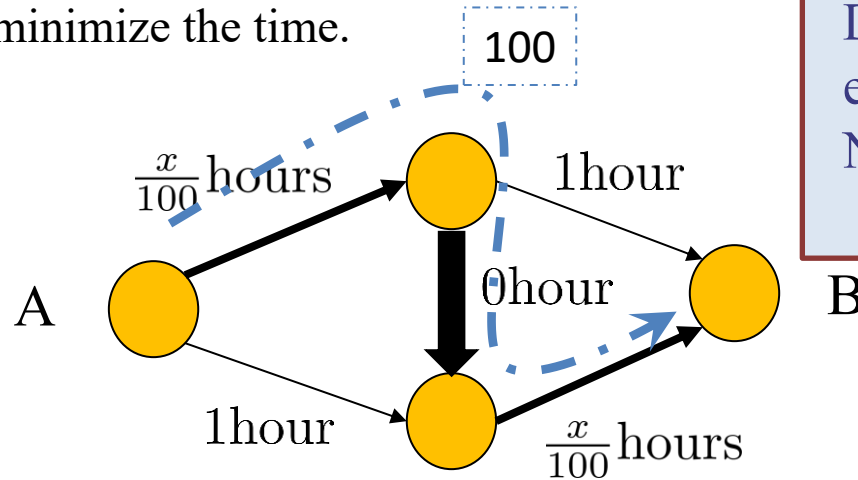
Suppose 100 drivers commute from A to B.
Drivers want to minimize the time.

Question: What if we **add** a new link?



Price of Anarchy

Suppose 100 drivers commute from A to B.
Drivers want to minimize the time.

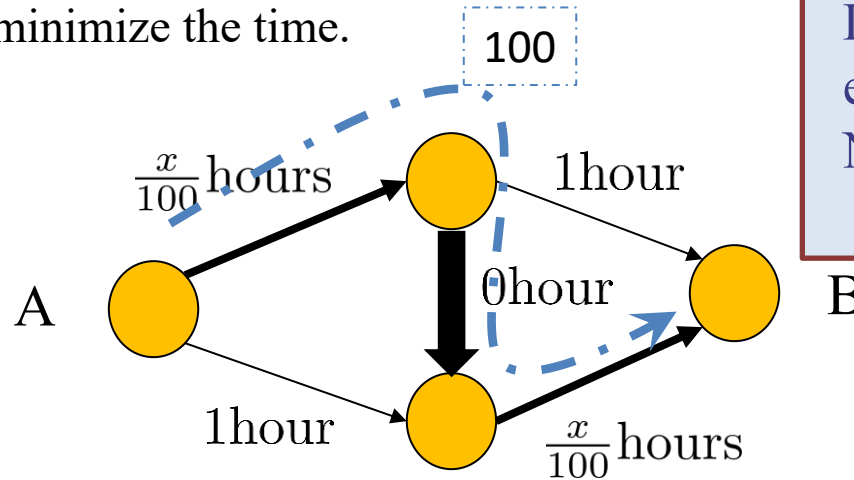


Delay is now 2 hours for everybody at the unique Nash equilibrium.
Braess's paradox

Adding a fast link is not always a good idea!

Price of Anarchy

Suppose 100 drivers commute from A to B.
Drivers want to minimize the time.



Delay is now 2 hours for everybody at the unique Nash equilibrium.
Braess's paradox

Adding a fast link is not always a good idea!

PoA = $\frac{\text{performance of worst case NE}}{\text{optimal performance if agents do not decide on their own}}$
Price of Anarchy (Koutsoupas, Papadimitriou 99').

4/3!!

Auctions

- Auctioneer has **one item** for sale.
- n **bidders** are interested in the item.
- Bidder i has **valuation** v_i for the item (unknown to Auctioneer).
- Each bidder i places a **bid** b_i , and based on $b_1, \dots, b_n$ auctioneer decides who gets the item and how much to **pay**.
- If bidder i gets the item and pays price p , her utility is $v_i - p$ otherwise 0.

Goal: Auctioneer wants to maximize her revenue! What is the correct pricing?
Who will get the item?