L01 Introduction

CS 295 Introduction to Algorithmic Game Theory
Ioannis Panageas
Course material

We will use canvas and material will also be posted on https://panageas.github.io/agt2021/

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

• Participation: 5%
• Homework: 30%
  – There will be given 2 Homeworks to solve (\textit{Latex}!).
• Scribing lecture notes: 30%
• Research Project/Present paper: 35%
  – Group of \(~3\). Report Deadline on 7\(^{th}\) of December.
  – Presentation last week of classes.
What is Game Theory?

Markets - Auctions

Routing

Evolution

Elections

Intro to AGT
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.

\begin{tabular}{|c|c|c|}
\hline
 & Deny & Confess \\
\hline
Deny & 1, 1 & 3, 0 \\
\hline
Confess & 0, 3 & 2, 2 \\
\hline
\end{tabular}
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.

<table>
<thead>
<tr>
<th></th>
<th>Deny</th>
<th>Confess</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deny</td>
<td>1, 1</td>
<td>3, 0</td>
</tr>
<tr>
<td>Confess</td>
<td>0, 3</td>
<td>2, 2</td>
</tr>
</tbody>
</table>
Example: Rock-Paper-Scissors

<table>
<thead>
<tr>
<th></th>
<th>Rock</th>
<th>Paper</th>
<th>Scissors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rock</td>
<td>0, 0</td>
<td>-1, 1</td>
<td>1, -1</td>
</tr>
<tr>
<td>Paper</td>
<td>1, -1</td>
<td>0, 0</td>
<td>-1, 1</td>
</tr>
<tr>
<td>Scissors</td>
<td>-1, 1</td>
<td>1, -1</td>
<td>0, 0</td>
</tr>
</tbody>
</table>
Example: Rock-Paper-Scissors

<table>
<thead>
<tr>
<th></th>
<th>Rock</th>
<th>Paper</th>
<th>Scissors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rock</td>
<td>0, 0</td>
<td>-1, 1</td>
<td>1, -1</td>
</tr>
<tr>
<td>Paper</td>
<td>1, -1</td>
<td>0, 0</td>
<td>-1, 1</td>
</tr>
<tr>
<td>Scissors</td>
<td>-1, 1</td>
<td>1, -1</td>
<td>0, 0</td>
</tr>
</tbody>
</table>

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 \( n \times m \).

\[ G = (R_{n \times m}, C_{n \times m}) \]

**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 $x^\top C y$. 

Intro to AGT
Bimatrix Games

Column player chooses $y \in \Delta_m$

Row player chooses $x \in \Delta_n$

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

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

$$x^* \top R y^* \geq x' \top R y^*$$

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

$$x^* \top C y^* \geq x^* \top C y'.$$
Example: Rock-Paper-Scissors

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

Remark:

Contrary to Prisoner’s Dilemma, in RPS \textit{randomization is necessary} for Nash equilibrium to exist!
Example: Rock-Paper-Scissors

<table>
<thead>
<tr>
<th>Rock</th>
<th>Paper</th>
<th>Scissors</th>
</tr>
</thead>
<tbody>
<tr>
<td>0, 0</td>
<td>-1, 1</td>
<td>1, -1</td>
</tr>
<tr>
<td>1, -1</td>
<td>0, 0</td>
<td>-1, 1</td>
</tr>
<tr>
<td>-1, 1</td>
<td>1, -1</td>
<td>0, 0</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th></th>
<th>Rock</th>
<th>Paper</th>
<th>Scissors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rock</td>
<td>0, 0</td>
<td>-1, 1</td>
<td>1, -1</td>
</tr>
<tr>
<td>Paper</td>
<td>1, -1</td>
<td>0, 0</td>
<td>-1, 1</td>
</tr>
<tr>
<td>Scissors</td>
<td>1, -1</td>
<td>1, -1</td>
<td>0, 0</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th></th>
<th>Rock</th>
<th>Paper</th>
<th>Scissors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rock</td>
<td>0, 0</td>
<td>-1, 1</td>
<td>2, -1</td>
</tr>
<tr>
<td>Paper</td>
<td>1, -1</td>
<td>0, 0</td>
<td>-1, 1</td>
</tr>
<tr>
<td>Scissors</td>
<td>-2, 1</td>
<td>1, -1</td>
<td>0, 0</td>
</tr>
</tbody>
</table>

Intro to AGT
Example: Modified Rock-Paper-Scissors

<table>
<thead>
<tr>
<th></th>
<th>Rock</th>
<th>Paper</th>
<th>Scissors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rock</td>
<td>0, 0</td>
<td>-1, 1</td>
<td>2, -1</td>
</tr>
<tr>
<td>Paper</td>
<td>1, -1</td>
<td>0, 0</td>
<td>-1, 1</td>
</tr>
<tr>
<td>Scissors</td>
<td>-2, 1</td>
<td>1, -1</td>
<td>0, 0</td>
</tr>
</tbody>
</table>

Not zero sum anymore!

Intro to AGT
**Example: Modified Rock-Paper-Scissors**

<table>
<thead>
<tr>
<th></th>
<th>25%</th>
<th>50%</th>
<th>25%</th>
</tr>
</thead>
<tbody>
<tr>
<td>🗼昭</td>
<td>0, 0</td>
<td>-1, 1</td>
<td>2,-1</td>
</tr>
<tr>
<td>🖌做</td>
<td>1,-1</td>
<td>0, 0</td>
<td>-1, 1</td>
</tr>
<tr>
<td>🎟玩</td>
<td>-2, 1</td>
<td>1, -1</td>
<td>0, 0</td>
</tr>
</tbody>
</table>

John Nash ’51:
There always **exists** a **Nash equilibrium** (finite games)!

Intro to AGT
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!
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.
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!
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.

Adding a fast link is not always a good idea!

$$\text{PoA} = \frac{\text{performance of worst case NE}}{\text{optimal performance if agents do not decide on their own}}$$

Price of Anarchy (Koutsoupias, Papadimitriou 99').
Auctions

• Auctioneer has one item for sale.
• *n* bidders are interested in the item.
• Bidder *i* has valuation *v*<sub>i</sub> for the item (unknown to Auctioneer).
• Each bidder *i* places a bid *b*<sub>i</sub>, and based on *b*<sub>1</sub>, ..., *b*<sub>n</sub> 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*<sub>i</sub> − *p* otherwise 0.

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