Lecture 16
NE Computation, PPAD and other TFNP classes

CS580

Ruta Mehta

Most slides are borrowed from Prof. C. Daskalakis’s presentation.
Menu

Existence Theorems: Nash, Brouwer, Sperner
Games

Players

Payoffs

Strategies

Randomize!
Nash (1950):
There exists a (stable) state where no player gains by unilateral deviation.

Nash equilibrium (NE)
### Games and Equilibria

#### Nash ’50:
An equilibrium exists in every game.

No poly-time algorithm known, despite intense effort.

#### Equilibrium:
A pair of randomized strategies so that no player has incentive to deviate if the other stays put.

<table>
<thead>
<tr>
<th></th>
<th>Kick</th>
<th>Dive</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Left</td>
<td>Right</td>
</tr>
<tr>
<td>1/2</td>
<td>2/5</td>
<td>3/5</td>
</tr>
<tr>
<td>Left</td>
<td>2, -1</td>
<td>-1, 1</td>
</tr>
<tr>
<td>Right</td>
<td>-1, 1</td>
<td>1, -1</td>
</tr>
</tbody>
</table>
Menu

Existence Theorems: Nash, Brouwer, Sperner
Brouwer’s Fixed Point Theorem

[Brouwer 1910]: Let \( f: D \to D \) be a continuous function from a convex and compact subset \( D \) of the Euclidean space to itself.

Then there exists an \( x \in D \) s.t. \( x = f(x) \).

Closed and bounded

A few examples, when \( D \) is the 2-dimensional disk.
Brouwer’s Fixed Point Theorem
Brouwer’s Fixed Point Theorem
Brouwer’s Fixed Point Theorem
Brouwer $\implies$ Nash
(Nash’51)
Nash’s Proof:

Lemma. If $x' = x$ then $x$ is best for Alice against $y$

$\forall i = 1, \ldots, m, \delta_i = \max\{(Ay)_i - x^T Ay, 0\}$,

$\forall i, \quad x'_i = \frac{x_i + \delta_i}{\sum_k(x_k + \delta_k)}$

$\equiv$ If $x^T Ay < z^T Ay$ for some $z \in \Delta_m$ then $x' \neq x$. 
Lemma. If $x^T Ay < z^T Ay$ for some $z \in \Delta_m$ then $x' \neq x$.

Proof:

\[
\begin{align*}
\forall i, & \quad \delta_i = \max\{(Ay)_i - x^T Ay, 0\}, \\
\forall i, & \quad x'_i = \frac{x_i + \delta_i}{\sum_k x_k + \delta_k}
\end{align*}
\]

\[k^* = \arg\max_{i=1}^m ((Ay)_i - x^T Ay)\]

\[k^* = \arg\min_{i: x_i > 0} (Ay)_i\]

\[\delta_{k^*} > 0\]

\[\delta_k = (Ay)_k - x^T Ay, 0 \geq 0\]

Claim: $x_{k^*}^1 \neq x_{k^*}$

\[x_{k^*}^1 = \frac{x_k^- + 0}{\sum_{i=1}^m x_i + \sum_{i=1}^n \delta_i^+} < x_{k^*}^-\]
Nash’s Proof

\[ f: \Delta_m \times \Delta_n \to \Delta_m \times \Delta_n , \quad (x', y') = f(x, y) \]

\[ \forall j, \quad \tau_j = \max \left\{ (x^T B)_j - x^T B y, 0 \right\}, \]

\[ \forall j, \quad y'_j = \frac{y_j + \tau_j}{\sum_k y_k + \tau_k} \]

**Lemma.** If \( y' = y \) then \( y \) is best for Bob against \( x \)

\[ \equiv \text{If} \quad x^T B y < x^T B z \quad \text{for some} \quad z \in \Delta_n \quad \text{then} \quad y' \neq y. \]
Visualizing Nash’s Proof

Penalty Shot Game

<table>
<thead>
<tr>
<th>Kick</th>
<th>Left</th>
<th>Right</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dive</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Left</td>
<td>1, -1</td>
<td>-1, 1</td>
</tr>
<tr>
<td>Right</td>
<td>-1, 1</td>
<td>1, -1</td>
</tr>
</tbody>
</table>

\[ f: [0,1]^2 \rightarrow [0,1]^2, \text{ continuous such that fixed points } \equiv \text{ Nash eq.} \]
## Visualizing Nash’s Proof

### Penalty Shot Game

<table>
<thead>
<tr>
<th>Kick</th>
<th>Left</th>
<th>Right</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dive</td>
<td>1, -1</td>
<td>-1, 1</td>
</tr>
<tr>
<td>Left</td>
<td>-1, 1</td>
<td>1, -1</td>
</tr>
</tbody>
</table>

![Penalty Shot Game Diagram](image)
Visualizing Nash’s Proof

Penalty Shot Game

\[
f: \Delta_2 \times \Delta_2 \rightarrow \Delta_2 \times \Delta_2
\]

\[
f: [0,1]^2 \rightarrow [0,1]^2
\]
Penalty Shot Game

<table>
<thead>
<tr>
<th>Kick</th>
<th>Left</th>
<th>Right</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dive</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Left</td>
<td>1, -1</td>
<td>-1, 1</td>
</tr>
<tr>
<td>Right</td>
<td>-1, 1</td>
<td>1, -1</td>
</tr>
</tbody>
</table>

Visualizing Nash’s Proof
Visualizing Nash’s Proof

Penalty Shot Game

<table>
<thead>
<tr>
<th></th>
<th>Kick</th>
<th>Dive</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Left</td>
<td>Right</td>
</tr>
<tr>
<td>Left</td>
<td>1, -1</td>
<td>-1, 1</td>
</tr>
<tr>
<td>Right</td>
<td>-1, 1</td>
<td>1, -1</td>
</tr>
</tbody>
</table>

fixed point

Pr[Right]
Menu

Existence Theorems: Nash, Brouwer, Sperner
Sperner’s Lemma (2-d)
Sperner’s Lemma (2-d)

[Sperner 1928]: Color the boundary using three colors in a legal way.
[Sperner 1928]: Color the boundary using three colors in a legal way. No matter how the internal nodes are colored, there exists a tri-chromatic triangle. In fact an odd number of those.
Sperner’s Lemma (2-d)

[Sperner 1928]: Color the boundary using three colors in a legal way. No matter how the internal nodes are colored, there exists a tri-chromatic triangle. In fact an odd number of those.
Sperner $\Rightarrow$ Brouwer
Sperner $\Rightarrow$ Brouwer (High-Level)

Given $f : [0,1]^2 \to [0,1]^2$

1) For all $\epsilon > 0$, existence of approximate fixed point $|f(x) - x| < \epsilon$, can be shown via Sperner’s lemma.

2) Then let $\epsilon \to 0$

For 1): Triangulate $[0,1]^2$;
Sperner $\Rightarrow$ Brouwer (High-Level)

Given $f : [0,1]^2 \to [0,1]^2$

1) For all $\epsilon > 0$, existence of approximate fixed point $|f(x) - x| < \epsilon$, can be shown via Sperner's lemma.

2) Then let $\epsilon \to 0$

For 1): Triangulate $[0,1]^2$;

Color points according to the direction of $(f(x) - x)$;
Sperner $\Rightarrow$ Brouwer (High-Level)

Given $f : [0,1]^2 \rightarrow [0,1]^2$
1) For all $\epsilon > 0$, existence of approximate fixed point $|f(x)-x| < \epsilon$, can be shown via Sperner’s lemma.
2) Then let $\epsilon \rightarrow 0$

For 1): Triangulate $[0,1]^2$;

Color points according to the direction of $(f(x)-x);$
Given \( f : [0,1]^2 \to [0,1]^2 \)

1) For all \( \epsilon > 0 \), existence of approximate fixed point \( |f(x) - x| < \epsilon \), can be shown via Sperner’s lemma.

2) Then let \( \epsilon \to 0 \)

For 1): Triangulate \([0,1]^2\);

Color points according to the direction of \((f(x) - x)\);

Apply Sperner.

**Sperner \(\Rightarrow\) Brouwer (High-Level)**
2D-Brouwer on the Square

Suppose \( f: [0,1]^2 \to [0,1]^2 \), continuous

\[
\forall \epsilon, \exists \delta(\epsilon) > 0, \text{ s.t. } d(x,y) < \delta(\epsilon) \Rightarrow d(f(x), f(y)) < \epsilon
\]

(by the Heine-Cantor theorem)

Choose small enough grid size so that...

**Claim:** If \( z \) a corner of a trichromatic triangle, then

Choosing \( \delta = \min\{\delta(\epsilon), \epsilon\} \)

\[
|f(z) - z|_\infty < c\delta, \quad c > 0
\]
Menu

Existence Theorems: Nash, Brouwer, Sperner

(Constructive) proof of Sperner $\rightarrow$ PPAD.
Proof of Sperner’s Lemma

[Sperner 1928]: Color the boundary using three colors in a legal way. No matter how the internal nodes are colored, there exists a tri-chromatic triangle. In fact an odd number of those.
Proof of Sperner’s Lemma

[Sperner 1928]: Color the boundary using three colors in a legal way. No matter how the internal nodes are colored, there exists a tri-chromatic triangle. In fact an odd number of those.
Proof of Sperner’s Lemma

Transition Rule: If $\exists$ red - yellow door cross it with red on your left hand.

[Sperner 1928]: Color the boundary using three colors in a legal way. No matter how the internal nodes are colored, there exists a tri-chromatic triangle. In fact an odd number of those.
Proof of Sperner’s Lemma

[Sperner 1928]: Color the boundary using three colors in a legal way. No matter how the internal nodes are colored, there exists a tri-chromatic triangle. In fact an odd number of those.
Proof Structure: A directed parity argument

Vertices of Graph \equiv Triangles
all vertices have in-degree, out-degree \leq 1

Proof:
\exists at least one trichromatic (artificial one) \ \Rightarrow \ \exists another trichromatic
Recall: Lemke-Howson Structure for 2-Nash

Vertices of Graph \( \equiv \) vertices of the polyhedron.

Proof: 0 fully-labeled \( \Rightarrow \exists \) another fully-labeled
The PPAD Class [Papadimitriou ’94] (Polynomial Parity Argument for Directed Graph)

Suppose that an exponentially large graph with vertex set \( \{0,1\}^n \) is defined by two circuits:

\[
\begin{align*}
N(v_1) = v_2 & \land P(v_2) = v_1 \\
\text{possible next} & \\
N & \\
\text{possible previous} & \\
P & \\
\end{align*}
\]

**END OF THE LINE:** \( P \) and \( N \) are given. If \( 0^n \) is an unbalanced node, find another unbalanced node. Otherwise output \( 0^n \).

\[
\text{PPAD} = \{ \text{Problems reducible to END OF A LINE} \} 
\]
END OF A LINE

\[
\text{\{0,1\}^n}
\]

\[
0^n
\]

\[
\Rightarrow P(v_2) = v_1 \text{ and } N(v_1) = v_2
\]

= solution
[Papadimitriou ’94]

PPAD-complete:
Nash eq. (even 2-player games),
Market eq., Sperner, Brouwer,
win-lose games, sparse games,
competitive eq. with equal income,
clearing payments in financial markets,
Fractional hypergraph matching,
Fractional stable path problem, …

$\varepsilon$-Brouwer: Given $f: D \to D$, find $x \in D$, s.t. $|f(x) - x| < \varepsilon$

$\varepsilon$-Nash: Profile from which no player can deviate and gains by more
than $\varepsilon$.

$\therefore$ Exact could be irrational
Menu

- Existence Theorems: Nash, Brouwer, Sperner
- (Constructive) proof of Sperner, and PPAD
- Why not use NP?
- Total Search problems.
NP, co-NP vs PPAD

Can they be NP-hard?  NO!

If there exists a solution?
- NP: poly-time verifier for YES answer
- co-NP: poly-time verifier for NO answer
- Here the answer is always YES!
  □ The problem is to find a solution.
Function NP (FNP): Search problems

Either find a solution or say there is none!

Problem $L \in \text{FNP}$ has a poly-time verifier $A_L$ s.t. $A_L(x, y) = 1$ if $y$ is a soln. of $x \in L$

It is *poly-time (Karp) reducible* to another problem $L' \in \text{FNP}$, associated with $A_{L'}$, iff there exist poly-time functions $f, g$ such that

(i) $f: \{0,1\}^* \rightarrow \{0,1\}^*$ maps inputs $x$ to $L$ into inputs $f(x)$ to $L'$

(ii) $\forall x,y: A_{L'}(f(x), y)=1 \implies A_{L}(x, g(y))=1$

$\forall x: A_{L'}(f(x), y)=0, \forall y \implies A_{L}(x, y)=0, \forall y$

A search problem $L'$ is *FNP-complete* iff

$L' \in \text{FNP}$

$\forall L \in \text{FNP}, L$ is poly-time reducible to $L'$.

SPERNER, NASH, BROUWER $\in \text{FNP}$. 

can’t reduce SAT to SPERNER, NASH or BROUWER

e.g. SAT
Total Function NP

Function NP (FNP): Search problems
   Either find a solution or say there is none!

Total FNP: A search problem is called total iff a solution is guaranteed.

PPAD \subseteq TFNP \subseteq (NP \cap \text{co-NP})

Complexity Theory of TFNP:

1. identify the combinatorial argument of existence, responsible for making these problems total;

2. define a complexity class inspired by the argument of existence;

3. make sure that the complexity of the problem was captured as tightly as possible (via completeness results).
PPAD:

In & out degree $\leq 1$

$0^n$ with in=0, out=1

$\exists$ another such node

$\bullet = \text{solution}$
Other arguments of existence, and resulting complexity classes

“If an undirected graph has a node of odd degree, then it must have another.”

“Every directed acyclic graph must have a sink.”

“If a function maps $n$ elements to $n-1$ elements, then there is a collision.”

Formally?
PPA: Polynomial Parity Argument
[Papadimitriou ’94]

“If a graph has a node of odd degree, then it must have another.”

Suppose that an exponentially large graph with vertex set \( \{0,1\}^n \) is defined by one circuit:

\[ A(v_1) \in \{ \text{node id}_1, \text{node id}_2 \} \]

**ODD DEGREE NODE**: Given \( C \), if \( 0^n \) has odd degree, find another node with odd degree. Otherwise say “yes”.

**PPA** = \( \{ \text{Search problems in FNP reducible to ODD DEGREE NODE} \} \)
The Undirected Graph

\{0,1\}^n

\[0^n\]

\[\ldots\]

\[\text{red circle} = \text{solution}\]
PLS: Polynomial Local Search
[Johnson, Papadimitriou, Yannakakis ’89]

“Every DAG has a sink.”

Suppose that a DAG with vertex set \(\{0,1\}^n\) is defined by two circuits:

\[
\begin{align*}
\text{node id} & \rightarrow A \rightarrow \{\text{node id}_1, \ldots, \text{node id}_k\} \\
\text{node id} & \rightarrow F \rightarrow \mathbb{R}
\end{align*}
\]

\(v_2 \in A(v_1) \quad \& \quad F(v_2) > F(v_1)\)

**FIND SINK.** Given \(C, F\): Find \(x\) s.t. \(F(x) \geq F(y)\), for all \(y \in C(x)\).

**PLS = \{ Search problems in FNP reducible to FIND SINK \}**
The DAG

$\{0,1\}^n$

$\bullet = \text{solution}$
PPP: Polynomial Pigeonhole Principle
[Papadimitriou ’94]

“If a function maps $n$ elements to $n-1$ elements, then there is a collision.”

Suppose that an exponentially large graph with vertex set $\{0,1\}^n$ is defined by one circuit:

\[
\text{COLLISION. Given } C: \text{ Find } x \text{ s.t. } C(x) = 0^n; \text{ or find } x \neq y \text{ s.t. } C(x) = C(y).
\]

\[
\text{PPP} = \{ \text{Search problems in FNP reducible to COLLISION} \}
\]