6.S890: Topics in Multiagent Learning

Lecture 8 – Prof. Daskalakis
Fall 2023
Refresher

• Last few times we stated and proved the theorems of Nash, Sperner, and Brouwer.

• [Nash’1950]: Every finite game has a Nash equilibrium.

• [Brouwer’1911]: Every continuous function \( f: D \to D \) from a convex compact set \( D \) to itself has a fixed point \( x^* = f(x^*) \).

• [Sperner’1928]: Every legal 3-coloring of a 2-d triangulated square has a tri-chromatic triangle.

• We also saw that:
  
  \[ \text{Sperner Lemma} \to \text{Brouwer Theorem} \to \text{Nash Theorem} \]

• which implies as a corollary that:

  \[ \text{Computing Nash Equilibria} \to \text{Computing Brouwer Fixed Points} \to \text{Finding Sperner Triangles} \]

• But what is the complexity of these problems?
  
  • we remarked that these problems are in the complexity class TFNP of \textit{total search problems in NP}.
  
  • “total”: they always have a solution, unlike e.g. SAT

• So what is their complexity?
100-feet overview of our methodology:

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).
OK, so what is the combinatorial argument of existence underlying Sperner, Brouwer and Nash?

A parity lemma in directed graphs:

A directed graph with an unbalanced node (a node with \text{indegree} \neq \text{outdegree}) must have another.
The PPAD Class [Papadimitriou ’94]
a complexity class capturing TFNP problems whose totality
is due to the directed parity argument

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

**End of the Line**: Given $P$ and $N$: If $0^n$ is an unbalanced node, find another unbalanced node. Otherwise output $0^n$.

$PPAD = \{ Search \ problems \ in \ FNP \ reducible \ to \ End \ of \ the \ line \}$
END OF THE LINE

\[ n \]

\[ 0^n \]

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

\[ v_1 \quad \Leftarrow \quad P(v_2) = v_1 \land N(v_1) = v_2 \]

\[ v_2 \]
Believed Location of PPAD
Litmus Test: Are NASH, BROUWER, and SPERNER PPAD-complete?

Partial Success: NASH, BROUWER, and SPERNER are in PPAD

→ means poly-time reduction
Poly-time Reductions that we just established:

[Polynomial-time Reductions]

[Polytime Reductions that we just established]
Menu

Refresher: Nash, Sperner, Brouwer, PPAD
Total Search Problems in NP
PPAD
PPAD-hardness of NASH
Menu

Refresher: Nash, Sperner, Brouwer, PPAD
Total Search Problems in NP
PPAD
PPAD-hardness of NASH
PPAD-Completeness of NASH

[Daskalakis, Goldberg, Papadimitriou'06]
PPAD-Completeness of NASH

Generic PPAD

Embed PPAD graph in $[0,1]^2$

2D-SPERNER

2D-BROUWER

ARITHMCIRCUITSAT

NASH
PPAD-Completeness of NASH

[Daskalakis, Goldberg, Papadimitriou’06]

Embed PPAD graph in $[0,1]^2$

2D-SPERNER  2D-BROUWER  ARITHM\textsc{CircuitSAT}  NASH
PPAD-Completeness of NASH

[Daskalakis, Goldberg, Papadimitriou’06]

Embed PPAD graph in $[0,1]^2$

Generic PPAD

2D-SPERNER

2D-BROUWER

ARITHM\text{CIRCUIT}SAT

NASH
**ARITHMCIRCUITSAT** [Daskalakis, Goldberg, Papadimitriou’06]

**INPUT:** A circuit comprising:

- variable nodes $v_1,\ldots, v_n$
- gate nodes $g_1,\ldots, g_m$ of types: $=\ , +\ , -\ , a\ , \times a\ , >$
- directed edges connecting variables to gates and gates to variables (loops are allowed);
- variable nodes have in-degree 1; gates have 0, 1, or 2 inputs depending on type as above; gates & nodes have arbitrary fan-out

**OUTPUT:** Values $v_1,\ldots, v_n \in [0,1]$ satisfying the gate constraints:

- assignment: $y == x_1$
- addition: $y == \min\{1, x_1 + x_2\}$
- subtraction: $y == \max\{0, x_1 - x_2\}$
- set equal to a constant: $y == \max\{0, \min\{1, a\}\}$
- multiply by constant: $y == \max\{0, \min\{1, a \cdot x_1\}\}$
Comparator Gate Constraints

\[ y = \begin{cases} 
1, & \text{if } x_1 > x_2 \\
0, & \text{if } x_1 < x_2 \\
*, & \text{if } x_1 = x_2 
\end{cases} \]

any value is allowed
ARITHMCIRCUITSAT (example)

Satisfying assignment?

\[ a = b = c = \frac{1}{2} \]
\textbf{ARITHM CIRCUIT SAT} \cite{DaskalakisGoldbergPapadimitriou06}

\textbf{INPUT:} A circuit comprising:
- variable nodes \(v_1, \ldots, v_n\)
- gate nodes \(g_1, \ldots, g_m\) of types: \(\oplus, \ominus, a, xa, >\)
- directed edges connecting variables to gates and gates to variables (loops are allowed);
- variable nodes have in-degree 1; gates have 0, 1, or 2 inputs depending on type as above; gates & nodes have arbitrary fanout.

\textbf{OUTPUT:} An assignment of values \(v_1, \ldots, v_n \in [0,1]\) satisfying:

\begin{align*}
\oplus: & \quad y \equiv x_1 \\
\ominus: & \quad y \equiv \min\{1, x_1 + x_2\} \\
\oplus: & \quad y \equiv \max\{0, x_1 - x_2\} \\
\oplus: & \quad y \equiv \max\{0, \min\{1, a\}\} \quad \text{[DGP'06]: but is PPAD-complete to find} \\
\oplus: & \quad y \equiv \max\{0, \min\{1, a \cdot x_1\}\} \\
\oplus: & \quad y = \begin{cases} 
1, & \text{if } x_1 > x_2 \\
0, & \text{if } x_1 < x_2 \\
*, & \text{if } x_1 = x_2 
\end{cases} \quad \text{[DGP'06]: Always exists satisfying assignment!}
\end{align*}
PPAD-Completeness of NASH

[Daskalakis, Goldberg, Papadimitriou’06]
**Game Gadgets:** Small games performing real arithmetic at their Nash equilibrium.
Addition Gadget

Suppose two strategies per player: \{0,1\}
then mixed strategy \equiv \text{a number in } [0,1] \text{ (the probability of playing 1)}

\text{e.g. } \textit{addition game}

\text{Claim: } \text{In any Nash equilibrium of a game containing the above gadget}
$$\Pr[z : 1] = \min\{\Pr[x : 1] + \Pr[y : 1], 1\}$$
all players have strategy set \( B = \{0,1\} \)

* player \( x,y \)'s payoff depends on other players' strategies
  * Player \( w \)'s payoff:
    - if \( w \) plays 0, \( w \)'s payoff doesn't depend on \( x,y \) strategy
      
      \[
      \begin{array}{c|c|c}
      x & y & w \\
      \hline
      0 & 0 & 1 \\
      1 & 1 & 0 \\
      \end{array}
      \]
    - if \( w \) plays 1, \( w \)'s payoff doesn't depend on \( x,y \) strategy
      
      \[
      \begin{array}{c|c|c}
      x & y & w \\
      \hline
      0 & 0 & 1 \\
      1 & 1 & 0 \\
      \end{array}
      \]
  
  \[
  z = \frac{x}{y}
  \]
  * Player \( z \)'s payoff:
    
    \[
    \begin{array}{c|c|c}
    x & y & w \\
    \hline
    0 & 0 & 0 \\
    1 & 1 & 0 \\
    \end{array}
    \]

Claim: In all Nash Eq of bigger game containing this, it must be that \( \Pr[z \text{ plays } 0] = \min \{ \Pr[x=1]+Pr[y=1], 1 \} \). 

Proof: Suppose \( \Pr[z=1] < \min \{ \Pr[x=1]+Pr[y=1], 1 \} < \Pr[x=1]+Pr[y=1] \)

\[
\Rightarrow \Pr[w=0]=1 \Rightarrow \Pr[z=1]=1 \quad \text{ contradiction}
\]

Suppose \( \Pr[z=1] > \min \{ \Pr[x=1]+Pr[y=1], 1 \} \)

well \( \min \{ \Pr[x=1]+Pr[y=1], 1 \} \) cannot be 1 as \( w \) \( \Pr[z=1] \geq 1 \text{ impossible} \)

Thus \( \Pr[z=1] > \Pr[x=1]+Pr[y=1] \)

\[
\Rightarrow \Pr[w=1]=1 \Rightarrow \Pr[z=1]=0 \quad \text{ contradiction}
\]

Thus only remaining possibility: \( \Pr[z=1] = \min \{ \Pr[x=1]+Pr[y=1], 1 \} \).
Subtraction Gadget

Suppose two strategies per player: \{0,1\}

then mixed strategy = a number in [0,1] (the probability of playing 1)

e.g. subtraction

Claim: In any Nash equilibrium of a game containing the above gadget

\[ u(w : 0) = \Pr[x : 1] - \Pr[y : 1] \]
\[ u(w : 1) = \Pr[z : 1] \]
\[ u(z : 0) = 0.5 \]
\[ u(z : 1) = 1 - \Pr[w : 1] \]
Notational convention: Use the name of the player and the probability of that player playing 1 interchangeably.

\[ x \xrightarrow{\text{Pr}[x : 1]} \]
List of Game Gadgets

<table>
<thead>
<tr>
<th>Operation</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>copy</td>
<td>$z = x$</td>
</tr>
<tr>
<td>addition</td>
<td>$z = \min{1, x + y}$</td>
</tr>
<tr>
<td>subtraction</td>
<td>$z = \max{0, x - y}$</td>
</tr>
<tr>
<td>set equal to a constant</td>
<td>$z = \max{0, \min{1, \alpha}}$</td>
</tr>
<tr>
<td>multiply by constant</td>
<td>$z = \max{0, \min{1, \alpha \cdot x}}$</td>
</tr>
<tr>
<td>comparison</td>
<td>$z = \begin{cases} 1, &amp; \text{if } x &gt; y \ 0, &amp; \text{if } x &lt; y \ *, &amp; \text{if } x = y \end{cases}$</td>
</tr>
</tbody>
</table>

$z$: “output player” of the gadget
$x, y$: “input players” of the gadget

If any of these gadgets is contained in a bigger game, these conditions hold at any Nash eq. of that bigger game.

Bigger game can only have edges into the “input players” and out of the “output players.”
Given arbitrary instance of ARITHMCIRCUITSAT can create multiplayer game by appropriately composing game gadgets corresponding to each of the gates.

At any Nash equilibrium of resulting game, the gate conditions are satisfied.
PPAD-Completeness of NASH

[Daskalakis, Goldberg, Papadimitriou’06]

DGP=Daskalakis-Goldberg-Papadimitriou

[Chen-Deng’06]

4-player Nash

3-player Nash

2-player Nash

MULTIPLAYERNASH

[DG’06]

ARITHMCCIRCUITSAT

[A]
PPAD-Completeness of NASH

[Daskalakis, Goldberg, Papadimitriou'06]
Poly-time Reductions that we just established:

[Daskalakis-Goldberg-Papadimitriou’06]:

PPAD
\rightarrow
SPERNER
\rightarrow
BROUWER
\rightarrow
NASH

PPAD
\rightarrow
SPERNER
\rightarrow
BROUWER
\rightarrow
NASH
Nash Equilibrium Complexity

[John Nash ’50]: A Nash equilibrium exists in every finite game.

Deep influence in Economics, enabling other existence results.

Proof non-constructive (uses Brouwer’s fixed point theorem)

No simpler proof has been discovered

[Daskalakis-Goldberg-Papadimitriou’06]: no simpler proof exists

i.e. Nash Equilibrium $\iff$ Brouwer’s Fixed Point Theorem
Menu

Refresher: Nash, Sperner, Brouwer, PPAD
Total Search Problems in NP
PPAD
PPAD-hardness of NASH
Final Musings
Menu

Refresher: Nash, Sperner, Brouwer, PPAD
Total Search Problems in NP
PPAD
PPAD-hardness of NASH
Final Musings
Other arguments of existence, and resulting complexity classes

“If a 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?
The Class PPA [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:

\[\text{ODDDEGREE\textsc{Node}:} \quad \text{Given } C: \text{If } 0^n \text{ has odd degree, find another node with odd degree. Otherwise output } 0^n.\]

\[\text{PPA} = \{ \text{Search problems in FNP reducible to ODDDEGREE\textsc{Node}} \}\]
ODD DEGREE NODE

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

\(= \) solution
Smith ∈ PPA

Smith: Given Hamiltonian cycle in 3-regular graph, find another one.

[Smith]: There must be another one.

Fig. 1. Smith's theorem in the case of a cubic graph.
The Class PLS [Johnson-Papadimitriou-Yannakakis ’89]

“Every DAG has a sink.”

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

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

\( v_2 \in C(v_1) \land F(v_2) > F(v_1) \)

**FINDSINK:** 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 FINDSINK* \}
FINDSINK

\{0,1\}^n

● = solution

Diagram: A graph with nodes connected by lines, indicating a network or communication pattern. The nodes are marked with red circles to indicate the solution set.
**LOCALMAXCUT** is PLS-complete

**LOCALMAXCUT:**
Given weighted graph $G = (V, E, w)$, find a partition $V = V_1 \cup V_2$ that is locally optimal (i.e. can’t move any single vertex to the other side to increase the cut size).

*[Schaffer-Yannakakis’91]: LocalMaxCut is PLS-complete.*
The Class PPP [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:

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

\[ \text{PPP} = \{ \text{Search problems in } FNP \text{ reducible to COLLISION} \} \]
CLS: continuous local search, capturing e.g. fixed points of gradient descent [Daskalakis-Papadimitriou’11]

CLS \equiv PPAD \cap PLS shown by [Fearnley-Goldber-Hollender-Savani’21]

In PPP: Factoring
PPP-complete: constrained Short-Integer-Solution

In PPA: Smith, Factoring
PPA-complete: consensus halving, fixed points
in unorientable spaces, combinatorial
nullstellensatz, chevalley-warning, Necklace
Splitting, Discrete Ham Sandwich