2/3/22: Hardness of Computing Nash Equilibria

Computational Problems:

- Nash: given game, compute a Nash equilibrium
- Two-Player Nash: given bimatrix game, compute a Nash equilibrium

Goal: Show that it is "unlikely" that there is a polynomial-time algorithm for Two-Player Nash

"unlikely": Would imply things that we don't think are true.
"Complexity-theoretic assumptions"

Too complicated/technical for this class, doesn't provide much new insight.

Instead: what does it mean?

Remember: NP

Definition 4.2.1 (NP) *A decision problem Q is in NP if there exists a polynomial time algorithm $V(I, X)$ such that*

1. *If I is a YES instance of Q then there exists some X such that $|X|$ is polynomial in $|I|$ and $V(I, X) = YES$*
2. *If I is a NO instance of Q then $V(I, X) = NO$ for all X*

NP-hard: Q is NP-hard if $\forall A \in NP$, there is a polynomial-time reduction from A to Q.

Decision version of Two-Player Nash: "is there a Nash"?

Always YES!

$\Rightarrow$ decision version not NP-hard

Side note: generalizations. (NRTV 2.2.2)

Thm: Following decision problems are NP-hard (even NP-complete):

- Are there ≥ 2 Nashes?
- Is there a Nash where Player 1's expected utility $\geq v$?
- Is there a Nash with support size $\geq v$?

$\vdots$

Attempt 1: Maybe non-decision version of Two-Player Nash is NP-hard?

Thm: If Two-Player Nash is NP-hard, then NP = coNP.

coNP: complement of NP. Can verify NO instances

Pf sketch: Need to show every problem in NP also in co-NP.

- Sufficient to show for any one NP-complete problem
- Let's show that SAT has short witnesses of NO that can be verified efficiently

Let φ instance of SAT

Since Two-Player Nash NP-hard (by assumption),

$$SAT \leq_p \text{Two-Player Nash}$$

$\Rightarrow$ Given φ, compute bimatrix game $A(\varphi)$ s.t.

if we can compute Nash (x,y) of $A(\varphi)$, can determine if φ is YES or NO

$\Rightarrow$ some alg $B(x,y)$ returns:

- YES if (x,y) Nash of $A(\varphi)$, φ YES
- NO if (x,y) Nash of $A(\varphi)$, φ NO

$\Rightarrow$ If φ a NO, witness is Nash (x,y) of $A(\varphi)$, can be verified by $B(x,y)$ ✓

Maybe not surprising: Two-Player Nash hard because search not because has to distinguish YES from NO.

Attempt 2: Define new complexity class

$$\text{TFNP} = \underbrace{\text{Total}}_{\text{Always some solution}}\ \underbrace{\text{Functional}}_{\text{search problems, not decision}}\ \text{NP}$$

Maybe Two-Player Nash is TFNP-complete?

Probably not: reasons to think that TFNP does not have any complete problems

(see Roughgarden 20.2.3)

Attempt 3: Think about Lemke-Howson more abstractly

High level, how did LH work?

- Both as proof that Nash exists, and algorithm?

1) There is a finite (possibly exponential) graph

2) Every vertex of graph has degree ≤ 2

3) There is one known source (vertex of degree 1), known as "standard source" (LH: $(0,0)$)

4) Every source other than standard source is a valid solution

5) Given a vertex, can determine its neighbors in graph in polynomial time

<u>Def</u> [informal, Papadimitriou '94]: PPAD is the class of search problems where existence of solution and algorithm to find it given by above properties

- PPAD = Polynomial Parity Argument (Directed)

Obviously Two-Player Nash is in PPAD.

So are many other important problems!

- Brouwer
- Sperner's Lemma

⋮

Why it was invented in first place!

<u>Q</u>: Every problem in PPAD has an exponential-time algorithm. Does every PPAD problem have a polytime algorithm?

Majority of complexity theorists: probably not

Thm: Two-Player Nash is PPAD-complete

Every problem in PPAD reduces to Two-Player Nash!

$\Rightarrow$ If there is a polytime algorithm for Two-Player Nash, then every problem in PPAD has a polytime algorithm

Nash's original proof is reduction of Nash to Brouwer: also a reduction in other direction!