Inapproximability of Nash equilibrium

Aviad Rubinstein
UC Berkeley

September 30, 2014
A short (algorithmic) history of Nash in two-player games
A short (algorithmic) history of Nash in two-player games

**Theorem ([CDT09, DGP09])**

*Finding a Nash equilibrium in two-player games is PPAD-complete.*
A short (algorithmic) history of Nash in two-player games

Theorem ([CDT09, DGP09])
Finding a Nash equilibrium in two-player games is PPAD-complete.

Well... but how about approximate Nash?
A short (algorithmic) history of Nash in two-player games

Theorem ([CDT09, DGP09])

Finding a Nash equilibrium in two-player games is PPAD-complete.

Well... but how about approximate Nash?

Theorem ([CDT09])

Finding an $n^\epsilon$-Nash equilibrium in two-player games is still PPAD-complete.
A short (algorithmic) history of Nash in two-player games

**Theorem ([CDT09, DGP09])**

Finding a Nash equilibrium in two-player games is PPAD-complete.

Well... but how about *approximate* Nash?

**Theorem ([CDT09])**

Finding an $n^\epsilon$-Nash equilibrium in two-player games is still PPAD-complete.

OK... but how about *constant approximation*?
OK... but how about *constant approximation*?
OK... but how about *constant approximation*?

**Theorem ([Das13])**

*Finding a multiplicative-\(\epsilon\)-Nash equilibrium in two-player games is still PPAD-complete.*
A short history ... (continued)

OK... but how about *constant approximation*?

**Theorem ([Das13])**

*Finding a multiplicative-\(\epsilon\)-Nash equilibrium in two-player games is still PPAD-complete.*

No, we mean *constant additive approximation*...
A short history … (continued)

OK… but how about constant approximation?

**Theorem ([Das13])**

*Finding a multiplicative-\(\epsilon\)-Nash equilibrium in two-player games is still PPAD-complete.*

No, we mean constant additive approximation…

**Theorem ([LMM03])**

*A quasi-polynomial time algorithm for finding \(\epsilon\)-Nash with constant number of players.*
OK... but how about constant approximation?

**Theorem ([Das13])**

*Finding a multiplicative-$\epsilon$-Nash equilibrium in two-player games is still PPAD-complete.*

No, we mean constant additive approximation...

**Theorem ([LMM03])**

*A quasi-polynomial time algorithm for finding $\epsilon$-Nash with constant number of players.*

Big open question: can we do better?
How do you like your multiplayer game?
Normal-form games

Fact

*Normal-form representation is exponential in the number of players*...
Normal-form games

Fact

Normal-form representation is exponential in the number of players...

(Curiously, there is no known algorithm that runs in time polynomial in this exponential representation. The current world record stands on $N^{\log \log \log N}$ [BBP14])
Value-query oracle

Given oracle access to the payoff tensor...
Value-query oracle

Given oracle access to the payoff tensor...

**Theorem ([HN13])**

*Exponential lower bound for deterministic algorithms.*
Value-query oracle

Given oracle access to the payoff tensor...

**Theorem ([HN13])**

*Exponential lower bound for deterministic algorithms.*

**Theorem ([Bab14])**

*Exponential lower bound for randomized algorithms.*
Succinct representations

Some classes of multiplayer games have a succinct representation:
Succinct representations

Some classes of multiplayer games have a succinct representation:

**Circuit**  Input is a (poly-size) circuit implementation of a value-query oracle (less standard).
Succinct representations

Some classes of multiplayer games have a succinct representation:

**Circuit** Input is a (poly-size) circuit implementation of a value-query oracle (less standard).

**Polymatrix** Each pair of players simultaneously plays a separate two-player game.
Succinct representations

Some classes of multiplayer games have a succinct representation:

**Circuit** Input is a (poly-size) circuit implementation of a value-query oracle (less standard).

**Polymatrix** Each pair of players simultaneously plays a separate two-player game.
(Every player has to play the same strategy in every two-player subgame,
Succinct representations

Some classes of multiplayer games have a succinct representation:

Circuit Input is a (poly-size) circuit implementation of a value-query oracle (less standard).

Polymatrix Each pair of players simultaneously plays a separate two-player game. (Every player has to play the same strategy in every two-player subgame, and her utility is the sum of her subgame utilities.)
Succinct representations

Some classes of multiplayer games have a succinct representation:

**Circuit**  Input is a (poly-size) circuit implementation of a value-query oracle (less standard).

**Polymatrix** Each pair of players simultaneously plays a separate two-player game.
(Every player has to play the same strategy in every two-player subgame, and her utility is the sum of her subgame utilities.)

**Graphical** The utility of each player depends only on the actions of a small number of “neighbors”.
Succinct representations

Some classes of multiplayer games have a succinct representation:

**Circuit** Input is a (poly-size) circuit implementation of a value-query oracle (less standard).

**Polymatrix** Each pair of players simultaneously plays a separate two-player game. (Every player has to play the same strategy in every two-player subgame, and her utility is the sum of her subgame utilities.)

**Graphical** The utility of each player depends only on the actions of a small number of “neighbors”.

**Main theorem**

*Degree 3, bipartite, polymatrix game where each player has 2 actions: $\epsilon$-approximate Nash is PPAD-complete.*
"free food!"
Two-player Bayesian Nash
Two-player Bayesian Nash

Definition (\(\epsilon\)-approximate Bayesian Nash equilibrium)

For every player \(i\) and type \(t_i\), mixed strategy \(x_i(t_i)\) is \(\epsilon\)-optimal, in expectation over other players' types (conditioned on \(t_i\)) & actions.

Corollary
Two-player game, constant number of actions: \(\epsilon\)-approximate Bayesian Nash equilibrium is PPAD-complete.
Two-player Bayesian Nash

Definition (\(\epsilon\)-approximate Bayesian Nash equilibrium)

For every player \(i\) and type \(t_i\), mixed strategy \(x_i(t_i)\) is \(\epsilon\)-optimal,
Two-player Bayesian Nash

Definition ($\epsilon$-approximate Bayesian Nash equilibrium)

For every player $i$ and type $t_i$, mixed strategy $x_i(t_i)$ is $\epsilon$-optimal, in expectation over other players’ types (conditioned on $t_i$) & actions.
Two-player Bayesian Nash

**Definition ($\epsilon$-approximate Bayesian Nash equilibrium)**

For every player $i$ and type $t_i$, mixed strategy $x_i(t_i)$ is $\epsilon$-optimal, in expectation over other players’ types (conditioned on $t_i$) & actions.

**Corollary**

*Two-player game, constant number of actions: $\epsilon$-approximate Bayesian Nash equilibrium is PPAD-complete.*
\( \epsilon \)-Generalized Circuit
"the 3SAT of PPAD"
Arithmetic Circuit
Generalized Circuit
Find an assignment $x : \mathbb{V} \to [0, 1]$ s.t.

$$\forall \text{ gate } (G, v_1, v_2, v) : x[v] = f_G(x[v_1] \pm \epsilon, x[v_2] \pm \epsilon) \pm \epsilon$$

**Corollary (Strengthening of [CDT09])**

$\epsilon$-GCircuit is PPAD-complete for constant $\epsilon > 0$. 

**Definition ($\epsilon$-GCircuit [CDT09])**
$\epsilon$-GCircuit

**Definition ($\epsilon$-GCircuit [CDT09])**

Find an assignment $x: V \rightarrow [0, 1]$ s.t. $\forall$ gate $(G, v_1, v_2, v)$
\[ \epsilon \text{-GCircuit} \]

**Definition (\( \epsilon \)-GCircuit [CDT09])**

Find an assignment \( x : V \rightarrow [0, 1] \) s.t. \( \forall \) gate \( (G, v_1, v_2, v) \)

\[ x[v] = f_G (x[v_1] \pm \epsilon, x[v_2] \pm \epsilon) \pm \epsilon \]
Definition ($\epsilon$-GCircuit [CDT09])

Find an assignment $x : V \rightarrow [0, 1]$ s.t. \( \forall \) gate \((G, v_1, v_2, v)\)

\[ x[v] = f_G(x[v_1] \pm \epsilon, x[v_2] \pm \epsilon) \pm \epsilon \]

Corollary (Strengthening of [CDT09])

$\epsilon$-GCircuit is PPAD-complete for constant $\epsilon > 0$. 
Non-monotone markets

Definition (Non-Monotone Market [CPY13])
Non-monotone markets

Definition (Non-Monotone Market [CPY13])

Raising the price of good $G$, while fixing all other prices, strictly increases the demand of $G$.
Non-monotone markets

Definition (Non-Monotone Market [CPY13])

Raising the price of good $G$, while fixing all other prices, strictly increases the demand of $G$

( Interesting to contrast non-monotonicity with weak gross substitutes... )
Non-monotone markets

**Definition (Non-Monotone Market [CPY13])**

Raising the price of good $G$, while fixing all other prices, strictly increases the demand of $G$.

(Interesting to contrast non-monotonicity with weak gross substitutes...)

**Definition (Weak Gross Substitutibility (WGS))**
Non-monotone markets

Definition (Non-Monotone Market [CPY13])

Raising the price of good $G$, while fixing all other prices, strictly increases the demand of $G$

( Interesting to contrast non-monotonicity with weak gross substitutes...

Definition (Weak Gross Substitutibility (WGS))

Raising the price of good $G$, while fixing all other prices, does not decrease the demand of other items.
Non-monotone markets

**Definition (Non-Monotone Market [CPY13])**

Raising the price of good $G$, while fixing all other prices, strictly *increases* the demand of $G$

( Interesting to contrast non-monotonicity with weak gross substitutes...

**Definition (Weak Gross Substitutibility (WGS))**

Raising the price of good $G$, while fixing all other prices, does not decrease the demand of other items.

For WGS, approximate market equilibrium can be found in poly-time [CMV05] )
PPAD-hardness for Non-monotone markets

Theorem ([CPY13])

For any family $\mathcal{U}$ of utilities that support non-monotone markets:
PPAD-hardness for Non-monotone markets

Theorem ([CPY13])

For any family $\mathcal{U}$ of utilities that support non-monotone markets:
If the utility of each bidder is either linear or from $\mathcal{U}$,
Corollary (Compare to [CPY13]'s result above)

If the utility of each bidder is either linear or from \( U \),
finding an \( \epsilon \)-tight approx market equilibrium is PPAD-hard.
Theorem ([CPY13])

For any family $\mathcal{U}$ of utilities that support non-monotone markets: If the utility of each bidder is either linear or from $\mathcal{U}$, finding an $1/n$-approximate market equilibrium is PPAD-hard.

Corollary (Compare to [CPY13]'s result above)

If the utility of each bidder is either linear or from $\mathcal{U}$,
**Theorem ([CPY13])**

For any family $\mathcal{U}$ of utilities that support non-monotone markets: If the utility of each bidder is either linear or from $\mathcal{U}$, finding an $1/n$-approximate market equilibrium is PPAD-hard.

**Corollary (Compare to [CPY13]'s result above)**

If the utility of each bidder is either linear or from $\mathcal{U}$, finding an $\epsilon$-tight approx market equilibrium is PPAD-hard.
Competitive Equilibrium from Equal Incomes

**Competitive Equilibrium from Equal Incomes (CEEI)**

- **Definitions**:
  - **M** goods (each with capacity **q**)
  - **N** bidders (each with linear valuation function **f**

\[ f_i(x_i) = \sum w_{i,j} x_{i,j} \]

- Competitive Equilibrium prices \{p_j\} that clear the market
- Equal Incomes: Each bidder receives 1 unit of budget
- Fails for many applications...
- Indivisible goods
- Complementarities / supplementarities
  - Examples: courses to students, shifts to workers, landing slots to airplanes, etc...

Competitive Equilibrium from Equal Incomes

Competitive Equilibrium from Equal Incomes (CEEI)

- $M$ goods
Competitive Equilibrium from Equal Incomes (CEEI)

- \( M \) goods (each with capacity \( q_j \));
Competitive Equilibrium from Equal Incomes

**Competitive Equilibrium from Equal Incomes (CEEI)**

- $M$ goods (each with capacity $q_j$);
- $N$ bidders
Competitive Equilibrium from Equal Incomes (CEEI)

- $M$ goods (each with capacity $q_j$);
- $N$ bidders
  (each with linear valuation function $f_i(x_i) = \sum w_{i,j} x_{i,j}$)
Competitive Equilibrium from Equal Incomes (CEEI)

- $M$ goods (each with capacity $q_j$);
- $N$ bidders
  (each with linear valuation function $f_i(x_i) = \sum w_{i,j} x_{i,j}$)

**Competitive Equilibrium** prices $\{p_j\}$ that clear the market.
Competitive Equilibrium from Equal Incomes (CEEI)

- \( M \) goods (each with capacity \( q_j \));
- \( N \) bidders
  (each with linear valuation function \( f_i(x_i) = \sum w_{i,j}x_{i,j} \))

**Competitive Equilibrium** prices \( \{p_j\} \) that clear the market

**Equal Incomes** Each bidder receives 1 unit of budget
### Competitive Equilibrium from Equal Incomes (CEEI)

- **$M$** goods (each with capacity $q_j$);
- **$N$** bidders (each with linear valuation function $f_i(x_i) = \sum w_{i,j} x_{i,j}$)

**Competitive Equilibrium** prices $\{p_j\}$ that clear the market

**Equal Incomes** Each bidder receives 1 unit of budget

---

<table>
<thead>
<tr>
<th>Fails for many applications...</th>
<th></th>
</tr>
</thead>
</table>
Competitive Equilibrium from Equal Incomes (CEEI)

- $M$ goods (each with capacity $q_j$);
- $N$ bidders
  (each with linear valuation function $f_i(x_i) = \sum w_{i,j}x_{i,j}$)

Competitive Equilibrium prices $\{p_j\}$ that clear the market

Equal Incomes Each bidder receives 1 unit of budget

Fails for many applications...

- Indivisible goods
Competitive Equilibrium from Equal Incomes

Competitive Equilibrium from Equal Incomes (CEEI)

- $M$ goods (each with capacity $q_j$);
- $N$ bidders
  (each with linear valuation function $f_i(x_i) = \sum w_{i,j}x_{i,j}$)

Competitive Equilibrium prices $\{p_j\}$ that clear the market

Equal Incomes Each bidder receives 1 unit of budget

Fails for many applications...

- Indivisible goods
- Complementarities / supplementarities
Competitive Equilibrium from Equal Incomes (CEEI)

- $M$ goods (each with capacity $q_j$);
- $N$ bidders (each with linear valuation function $f_i(x_i) = \sum w_{i,j}x_{i,j}$)

Competitive Equilibrium prices $\{p_j\}$ that clear the market

Equal Incomes Each bidder receives 1 unit of budget

Fails for many applications...

- Indivisible goods
- Complementarities / supplementarities
Competitive Equilibrium from Equal Incomes (CEEI)

- **$M$ goods** (each with capacity $q_j$);
- **$N$ bidders**
  (each with linear valuation function $f_i(x_i) = \sum w_{i,j} x_{i,j}$)

**Competitive Equilibrium** prices $\{p_j\}$ that clear the market

**Equal Incomes** Each bidder receives 1 unit of budget

Fails for many applications...

- Indivisible goods
- Complementarities / supplementarities

Examples: courses to students, shifts to workers, landing slots to airplanes, etc...
CourseMatch: equilibrium from equal incomes (A-CEEI)

Approximate Competitive Equilibrium from Equal Incomes (A-CEEI)

Indivisible goods (each with capacity $q_j$); $N$ bidders (each with arbitrary valuation function $f_i(x_i): 2^M \rightarrow \mathbb{R}$)

$\alpha$-Competitive Equilibrium prices $\{p_j\}$ that $\alpha$-clear the market

$\beta$-Equal Incomes Each bidder receives a budget $b_i \in [1, 1 + \beta]$

Theorem (Existence [Bud11])

For every $\beta > 0$, there exists a $(\alpha(M), \beta)$-CEEI.
Approximate Competitive Equilibrium from Equal Incomes (A-CEEI)

- $M$ indivisible goods (each with capacity $q_j$);
Approximate Competitive Equilibrium from Equal Incomes (A-CEEI)

- $M$ indivisible goods (each with capacity $q_j$);
- $N$ bidders
  (each with arbitrary valuation function $f_i(x_i) : 2^M \rightarrow \mathbb{R}$)

CourseMatch: equilibrium from equal incomes (A-CEEI)
CourseMatch: equilibrium from equal incomes (A-CEEI)

Approximate Competitive Equilibrium from Equal Incomes (A-CEEI)

- $M$ indivisible goods (each with capacity $q_j$);
- $N$ bidders
  (each with arbitrary valuation function $f_i(x_i) : 2^M \rightarrow \mathbb{R}$)

$\alpha$-Competitive Equilibrium prices $\{p_j\}$ that $\alpha$-clear the market
Approximate Competitive Equilibrium from Equal Incomes (A-CEEI)

- $M$ indivisible goods (each with capacity $q_j$);
- $N$ bidders (each with arbitrary valuation function $f_i(x_i) : 2^M \rightarrow \mathbb{R}$)

$\alpha$-Competitive Equilibrium prices $\{p_j\}$ that $\alpha$-clear the market

$\beta$-Equal Incomes Each bidder receives a budget $b_i \in [1, 1 + \beta]$
Approximate Competitive Equilibrium from Equal Incomes (A-CEEI)

- $M$ indivisible goods (each with capacity $q_j$);
- $N$ bidders (each with arbitrary valuation function $f_i(x_i) : 2^M \rightarrow \mathbb{R}$)

$\alpha$-Competitive Equilibrium prices $\{p_j\}$ that $\alpha$-clear the market

$\beta$-Equal Incomes Each bidder receives a budget $b_i \in [1, 1 + \beta]$
Approximate Competitive Equilibrium from Equal Incomes (A-CEEI)

- $M$ indivisible goods (each with capacity $q_j$);
- $N$ bidders (each with arbitrary valuation function $f_i(x_i) : 2^M \to \mathbb{R}$)

$\alpha$-Competitive Equilibrium prices $\{p_j\}$ that $\alpha$-clear the market

$\beta$-Equal Incomes Each bidder receives a budget $b_i \in [1, 1 + \beta]$

Theorem (Existence [Bud11])

For every $\beta > 0$, there exists a $(\alpha(M), \beta)$-CEEI.
Theorem (Existence [Bud11])

For every $\beta > 0$, there exists a $(\alpha(M), \beta)$-CEEI.
Theorem (Existence [Bud11])

For every $\beta > 0$, there exists a $(\alpha(M), \beta)$-CEEI.

CourseMatch: assigning classes to students with A-CEEI
Theorem (Existence [Bud11])

For every $\beta > 0$, there exists a $(\alpha(M), \beta)$-CEEI.

CourseMatch: assigning classes to students with A-CEEI

- Heuristic algorithm [OBS10]
CourseMatch: A-CEEI... (continued)

**Theorem (Existence [Bud11])**

For every $\beta > 0$, there exists a $(\alpha(M), \beta)$-CEEI.

**CourseMatch: assigning classes to students with A-CEEI**

- Heuristic algorithm [OBS10]
- Abe Othman: Implementation at Wharton - since Fall 2013...
CourseMatch: A-CEEI... (continued)

Theorem (Existence [Bud11])

For every $\beta > 0$, there exists a $(\alpha(M), \beta)$-CEEI.

CourseMatch: assigning classes to students with A-CEEI

- Heuristic algorithm [OBS10]
- Abe Othman: Implementation at Wharton - since Fall 2013...

Corollary (Strengthening of [OPR14])

Finding an $(\alpha(M), \beta)$-CEEI is PPAD-complete for constant $\beta > 0$
"It’s elementary, my dear Watson"
What is PPAD? [Pap94]

Definition

$G$ is a graph over \{$0, 1\}$

$g \in \text{Poly-time}$ functions $P, S$ give the predecessor and successor of each vertex $P(0^n) = \phi$

Find another vertex $v \neq 0^n$ such that $S(v) = \phi$ or $P(v) = \phi$
What is PPAD? [Pap94]

**Definition**

- $G$ is a graph over $\{0, 1\}^n$ (exponential size!)
What is PPAD? [Pap94]

Definition

- $G$ is a graph over $\{0, 1\}^n$ (exponential size!)
- $d_{in}, d_{out} \leq 1$
What is PPAD? [Pap94]

**Definition**

- $G$ is a graph over $\{0, 1\}^n$ (exponential size!)
  - $d_{in}, d_{out} \leq 1$
- Poly-time functions $P, S$ give the predecessor and successor of each vertex
What is PPAD? [Pap94]

**Definition**

- $G$ is a graph over $\{0, 1\}^n$ (exponential size!)
  - $d_{in}, d_{out} \leq 1$
- Poly-time functions $P, S$ give the predecessor and successor of each vertex
- $P(0^n) = \phi$
What is PPAD? [Pap94]

Definition

- $G$ is a graph over $\{0, 1\}^n$ (exponential size!)
  - $d_{in}, d_{out} \leq 1$
- Poly-time functions $P, S$ give the predecessor and successor of each vertex
- $P(0^n) = \phi$

Find another vertex $v \neq 0^n$ such that $S(v) = \phi$ or $P(v) = \phi$
Previous hardness results for Nash equilibrium [DGP09, CDT09, Bab14] follow the “DGP framework”: 

1. Show that finding a Brouwer fixed point is (PPAD-) hard; 
2. Reduce Brouwer to Nash. 
In particular, we need to find a way to embed a path following problem as a continuous function...
DGP framework

Previous hardness results for Nash equilibrium [DGP09, CDT09, Bab14] follow the “DGP framework”:

1. Show that finding a Brouwer fixed point is (PPAD-) hard;
DGP framework

Previous hardness results for Nash equilibrium [DGP09, CDT09, Bab14] follow the “DGP framework”:

1. Show that finding a Brouwer fixed point is (PPAD-) hard;
2. Reduce Brouwer to Nash.
DGP framework

Previous hardness results for Nash equilibrium [DGP09, CDT09, Bab14] follow the “DGP framework”:

1. Show that finding a Brouwer fixed point is (PPAD-) hard;
2. Reduce Brouwer to Nash.

In particular, we need to find a way to embed a path following problem as a continuous function...
The HPV construction

We use an embedding of a path due to [HPV89]:

Partition the \((n+1)\)-dimensional hypercube into smaller subcubes for constant hardness, use high dimension + constant side length. Define an (exponentially long) path between the centers of the subcubes. Embed the flow of a continuous function along the path so that the only fixed point corresponds to the end of the path. (For PPAD, need to modify to embed many paths.)
The HPV construction

We use an embedding of a path due to [HPV89]:

- Partition the \((n + 1)\)-dimensional hypercube into smaller subcubes
The HPV construction

We use an embedding of a path due to [HPV89]:

- Partition the \((n + 1)\)-dimensional hypercube into smaller subcubes
  - for constant hardness, use high dimension + constant side length
The HPV construction

We use an embedding of a path due to [HPV89]:

- Partition the \((n + 1)\)-dimensional hypercube into smaller subcubes
  - for constant hardness, use high dimension + constant side length
- Define an (exponentially long) path between the centers of the subcubes
The HPV construction

We use an embedding of a path due to [HPV89]:

- Partition the \((n + 1)\)-dimensional hypercube into smaller subcubes
  - for constant hardness, use high dimension + constant side length
- Define an (exponentially long) path between the centers of the subcubes
- Embed the flow of a continuous function along the path s.t. the only fixed point corresponds to the end of the path
The HPV construction

We use an embedding of a path due to [HPV89]:

- Partition the \((n + 1)\)-dimensional hypercube into smaller subcubes
  - for constant hardness, use high dimension + constant side length
- Define an (exponentially long) path between the centers of the subcubes
- Embed the flow of a continuous function along the path s.t. the only fixed point corresponds to the end of the path

(For PPAD, need to modify to embed many paths.)
The HPV construction (continued)

(Figure taken from [HPV89])
Averaging gadgets

We reduce from discrete (PPAD) to continuous (Brouwer and Nash), by matching subcubes ↔ vertices.
Averaging gadgets

We reduce from discrete (PPAD) to continuous (Brouwer and Nash), by matching subcubes $\leftrightarrow$ vertices.
What happens near the facets between the subcubes?
Averaging gadgets

We reduce from discrete (PPAD) to continuous (Brouwer and Nash), by matching subcubes $\leftrightarrow$ vertices.
What happens near the facets between the subcubes?

Averaging gadget of [DGP09]
We reduce from discrete (PPAD) to continuous (Brouwer and Nash), by matching subcubes ↔ vertices. What happens near the facets between the subcubes?

Averaging gadget of [DGP09]

- In constant dimension, we can smooth the function by averaging over a polynomial number of points in a ball around the input point.
We reduce from discrete (PPAD) to continuous (Brouwer and Nash), by matching subcubes $\leftrightarrow$ vertices.
What happens near the facets between the subcubes?

**Averaging gadget of [DGP09]**

- In constant dimension, we can smooth the function by averaging over a polynomial number of points in a ball around the input point.
- When we go to high dimensions, averaging over a ball requires exponentially many samples.
Averaging gadgets (continued)

We reduce from discrete (PPAD) to continuous (Brouwer and Nash), by matching subcubes ↔ vertices. What happens near the facets between the subcubes?
Averaging gadgets (continued)

We reduce from discrete (PPAD) to continuous (Brouwer and Nash), by matching subcubes ↔ vertices. What happens near the facets between the subcubes?

Averaging gadget of [CDT09]
We reduce from discrete (PPAD) to continuous (Brouwer and Nash), by matching subcubes ↔ vertices.
What happens near the facets between the subcubes?

Averaging gadget of [CDT09]

- Instead of sampling in every direction, it suffices to average over points on a single line (parallel to the 1 vector).
We reduce from discrete (PPAD) to continuous (Brouwer and Nash), by matching subcubes $\leftrightarrow$ vertices.
What happens near the facets between the subcubes?

**Averaging gadget of [CDT09]**

- Instead of sampling in every direction, it suffices to average over points on a single line (parallel to the $\mathbf{1}$ vector).
- Every point on the line may still be close to some facets - but if we sample $n^3$ points, most of them are far from every facet.
Averaging gadgets (continued)

We reduce from discrete (PPAD) to continuous (Brouwer and Nash), by matching subcubes $\leftrightarrow$ vertices.
What happens near the facets between the subcubes?

**Averaging gadget of [CDT09]**

- Instead of sampling in every direction, it suffices to average over points on a single line (parallel to the $1$ vector).

- Every point on the line may still be close to some facets - but if we sample $n^3$ points, most of them are far from every facet.

- But if we sample $n^3$ points on a line, they cannot be $\epsilon$-far from each other...
The corners kick

Key observation:
The corners kick

Key observation:

Fact

*In the HPV construction, all the corners look the same!*
The corners kick

Key observation:

Fact

In the HPV construction, all the corners look the same!

This paves the path to a reduction to $\epsilon$-Nash:
The corners kick

Key observation:

Fact

*In the HPV construction, all the corners look the same!*

This paves the path to a reduction to $\epsilon$-Nash:

1. Treat corners separately;
The corners kick

Key observation:

Fact

*In the HPV construction, all the corners look the same!*

This paves the path to a reduction to $\epsilon$-Nash:

1. Treat corners separately;
2. Average over a constant number of points.
The corners kick

Key observation:

**Fact**

*In the HPV construction, all the corners look the same!*

This paves the path to a reduction to $\epsilon$-Nash:

1. Treat corners separately;
2. Average over a constant number of points.
3. ...
Summary

Main theorem

Degree 3, bipartite, polymatrix game where each player has 2 actions: \( \epsilon \)-approximate Nash is PPAD-complete.
Summary

Main theorem

*Degree 3, bipartite, polymatrix game where each player has 2 actions: ϵ-approximate Nash is PPAD-complete.*

Corollaries
Summary

Main theorem

Degree 3, bipartite, polymatrix game where each player has 2 actions: $\epsilon$-approximate Nash is PPAD-complete.

Corollaries

- Bayesian two-player game
Summary

Main theorem

*Degree 3, bipartite, polymatrix game where each player has 2 actions: $\epsilon$-approximate Nash is PPAD-complete.*

Corollaries

- *Bayesian two-player game*
- *$\epsilon$-GCircuit*
Summary

Main theorem

*Degree 3, bipartite, polymatrix game where each player has 2 actions: $\epsilon$-approximate Nash is PPAD-complete.*

Corollaries

- *Bayesian two-player game*
- *$\epsilon$-GCircuit*
- *Non-monotone markets*
Summary

Main theorem

Degree 3, bipartite, polymatrix game where each player has 2 actions: $\epsilon$-approximate Nash is PPAD-complete.

Corollaries

- Bayesian two-player game
- $\epsilon$-GCircuit
- Non-monotone markets
- A-CEEI (CourseMatch)
Open questions

Main theorem

*Degree 3, bipartite, polymatrix game where each player has 2 actions: $\epsilon$-approximate Nash is PPAD-complete.*
Open questions

Main theorem

Degree 3, bipartite, polymatrix game where each player has 2 actions: $\epsilon$-approximate Nash is PPAD-complete.

- What about bimatrix games?
Main theorem

Degree 3, bipartite, polymatrix game where each player has 2 actions: $\epsilon$-approximate Nash is PPAD-complete.

- What about bimatrix games?
- More corollaries?
Open questions

Main theorem

Degree 3, bipartite, polymatrix game where each player has 2 actions: $\epsilon$-approximate Nash is PPAD-complete.

- What about bimatrix games?
- More corollaries? (other market equilibria?)


References II


References III

Constantinos Daskalakis.
On the complexity of approximating a nash equilibrium.

Constantinos Daskalakis, Paul W. Goldberg, and Christos H. Papadimitriou.
The complexity of computing a nash equilibrium.

Sergiu Hart and Noam Nisan.
The query complexity of correlated equilibria.

Exponential lower bounds for finding brouwer fix points.


Christos H. Papadimitriou.
On the complexity of the parity argument and other inefficient proofs of existence.