

Algorithmic Game Theory
Algorithms for normal-form games and
approximate Nash equilibria

Vangelis Markakis
markakis@gmail.com

Outline

- Algorithms for finding equilibria in general normal form games
 - The support theorem
 - Analysis of 2×2 and $2 \times n$ games
 - Complexity of general $n \times m$ games
- Approximate Nash equilibria
 - A subexponential algorithm for any constant $\epsilon > 0$
 - Polynomial time algorithms

Nash equilibria: Existence and computation

- In 0-sum games
 - von Neumann's theorem establishes both existence and an algorithm for finding an equilibrium
 - Boils down to solving one linear program
- In general games?
 - Nash's theorem guarantees only existence
 - Big research question over the last 25 years

The support of a strategy

- To come up with efficient algorithms, we need to understand better the properties of Nash equilibria
- **Definition:** For a mixed strategy $\mathbf{p} = (p_1, p_2, \dots, p_n)$, the *support* of \mathbf{p} is the set of pure strategies that have a positive probability of being selected, when we play \mathbf{p}

$$\text{Supp}(\mathbf{p}) = \{i: p_i > 0\}$$

- E.g. if $\mathbf{p} = (2/7, 0, 0, 3/7, 0, 2/7)$, then $\text{Supp}(\mathbf{p}) = \{1, 4, 6\}$
 - For pl. 1, $\text{Supp}(\mathbf{p})$ shows us which rows have a chance to be selected according to \mathbf{p}
 - Respectively, for a strategy of pl. 2, it shows the columns

Utility functions revisited

- Let (\mathbf{p}, \mathbf{q}) be a strategy profile in a $n \times m$ game
 - $\mathbf{p} = (p_1, p_2, \dots, p_n)$, $\mathbf{q} = (q_1, q_2, \dots, q_m)$
- Analyzing the utility function of pl. 1:

$$u_1(\mathbf{p}, \mathbf{q}) = \sum_{i=1}^n \sum_{j=1}^m p_i \cdot q_j \cdot u_1(s_i, t_j) = \sum_{i=1}^n p_i \sum_{j=1}^m q_j \cdot u_1(s_i, t_j) = \sum_{i=1}^n p_i \cdot u_1(e^i, \mathbf{q})$$

- The last term can also be written in terms of the support of \mathbf{p} , hence:

$$u_1(\mathbf{p}, \mathbf{q}) = \sum_{i \in \text{Supp}(\mathbf{p})} p_i \cdot u_1(e^i, \mathbf{q})$$

Support properties at Nash equilibria

- Let (\mathbf{p}, \mathbf{q}) be a Nash equilibrium and let $i, j \in \text{Supp}(\mathbf{p})$
 - $p_i > 0, p_j > 0$
- How are the quantities $u_1(e^i, \mathbf{q})$ and $u_1(e^j, \mathbf{q})$ related?
- If $u_1(e^i, \mathbf{q}) > u_1(e^j, \mathbf{q})$, then pl. 1 has an incentive to reduce the probability p_j and increase the probability p_i
 - But then (\mathbf{p}, \mathbf{q}) would not be a Nash equilibrium
 - Similarly, if $u_1(e^i, \mathbf{q}) < u_1(e^j, \mathbf{q})$
 - The only choice at an equilibrium is to have $u_1(e^i, \mathbf{q}) = u_1(e^j, \mathbf{q})$
- If $i \in \text{Supp}(\mathbf{p})$ and $j \notin \text{Supp}(\mathbf{p})$?
 - Then it must hold that $u_1(e^i, \mathbf{q}) \geq u_1(e^j, \mathbf{q})$, otherwise (\mathbf{p}, \mathbf{q}) is not an equilibrium

Support properties at Nash equilibria

Support theorem: A profile (\mathbf{p}, \mathbf{q}) is a Nash equilibrium if and only if

- i. $\forall i, j \in \text{Supp}(\mathbf{p}), u_1(e^i, \mathbf{q}) = u_1(e^j, \mathbf{q})$
- ii. $\forall i, j \in \text{Supp}(\mathbf{q}), u_2(\mathbf{p}, e^i) = u_2(\mathbf{p}, e^j)$
- iii. $\forall i \in \text{Supp}(\mathbf{p})$ and $\forall j \notin \text{Supp}(\mathbf{p}), u_1(e^i, \mathbf{q}) \geq u_1(e^j, \mathbf{q})$
- iv. $\forall i \in \text{Supp}(\mathbf{q})$ and $\forall j \notin \text{Supp}(\mathbf{q}), u_2(\mathbf{p}, e^i) \geq u_2(\mathbf{p}, e^j)$

Support properties at Nash equilibria

In other words:

- If a pure strategy is used with positive probability at a Nash equilibrium, then this strategy should be at least as good as any other pure strategy, **given** the other player's strategy
- 2 pure strategies that have positive probability at a Nash equilibrium must have the same utility, given the other player's strategy
- The theorem yields a way to check if a profile is a Nash equilibrium
- And helps us understand why some profiles cannot form an equilibrium

Example

Use the support theorem to check if the profile (\mathbf{p}, \mathbf{q}) with $\mathbf{p} = (3/4, 0, 1/4)$, $\mathbf{q} = (0, 1/3, 2/3)$ is an equilibrium in the following game

	t_1	t_2	t_3
s_1	1, 2	3, 3	1, 1
s_2	3, 2	0, 1	2, 5
s_3	2, 4	5, 1	0, 7

Finding Nash equilibria

Corollary: If we knew the support of the strategies in one equilibrium profile, then we could compute a Nash equilibrium in polynomial time

In other words: if we only knew which rows and columns are needed in an equilibrium, we could then compute the probabilities of the mixed strategies

Proof:

- Suppose that someone guesses the support for both players
- All the conditions of the support theorem are linear functions of $p_1, p_2, \dots, p_n, q_1, q_2, \dots, q_m$
- We would also need to add that $\sum_i p_i = 1, \sum_i q_i = 1$
- By solving a single linear program (or a system of linear inequalities) we can compute the probabilities of the mixed strategies

Finding Nash equilibria

- At the end, finding a Nash equilibrium is a combinatorial problem
- It suffices to find the right supports
- Brute-force algorithm:
 - Enumerate all possible pairs of supports for the two players
 - For each pair of supports, check if the corresponding linear program has a solution
- Complexity of brute-force in $n \times m$ games: prohibitive!
 - 2^n choices for pl. 1
 - 2^m choices for pl. 2
 - We need to run $O(2^{n+m})$ linear programs

Finding Nash equilibria

- Can we reduce it to solving only a few linear programs?
- Or a single LP?
- Probably no...
- **Note:** If the problem is solvable in polynomial time, then it can be reduced to a 0-sum game, by what we said in previous lecture
- It turns out that finding Nash equilibria is a special case of a “linear complementarity problem” [Cottle, Dantzig, 1960s]

Finding Nash equilibria

Linear Complementarity Problems (LCP)

- They arise in various contexts in Operations Research
- A class of non-linear programs
- Non-linear constraints for Nash equilibria:
 - By the support theorem, we need to express the fact that if $p_i > 0$ at an equilibrium, then the i -th pure strategy gives maximum payoff among all pure strategies
- We cannot express such “if” statements with a linear program
- Instead: let w be the expected payoff of pl. 1 at an equilibrium (\mathbf{p}, \mathbf{q})
- Support theorem \Rightarrow if $p_i > 0$, then $u_1(e^i, \mathbf{q}) = w$
- Equivalently: $p_i \cdot (u_1(e^i, \mathbf{q}) - w) = 0$ [complementarity condition]



Nash equilibria as a LCP

- Variables:
 - $p_1, p_2, \dots, p_n, q_1, q_2, \dots, q_m$: for the probabilities of the mixed strategies
 - w, w' : for the expected utilities of the 2 players
- Constraints:
 - $\sum_i p_i = 1, \sum_i q_i = 1$
 - $p_1 \geq 0, p_2 \geq 0, \dots, q_1 \geq 0, \dots, q_m \geq 0$
 - $w \geq u_1(e^i, \mathbf{q})$ for $i=1, \dots, n$
 - $w' \geq u_2(\mathbf{p}, e^j)$ for $j=1, \dots, m$
 - $p_i \cdot (u_1(e^i, \mathbf{q}) - w) = 0$, for $i=1, \dots, n$
 - $q_j \cdot (u_2(\mathbf{p}, e^j) - w') = 0$, for $j=1, \dots, m$
- Algorithm for solving LCPs: [Lemke, Howson '64]
 - Exponential time in worst case, but relatively ok on average
 - Based on ideas similar to simplex but for non-linear problems
 - see GAMBIT <http://www.gambit-project.org/>

Finding Nash equilibria

- So far, we have only seen exponential time algorithms...
- In what cases can the support theorem help us in having better algorithms?
- 2x2 games:
 - If there is a mixed strategy equilibrium then the support for pl. 1 must contain both rows
 - The support of pl. 2 must contain both columns
 - Applying the support theorem, it must hold that
$$u_1(e^1, \mathbf{q}) = u_1(e^2, \mathbf{q}), \text{ and } u_2(\mathbf{p}, e^1) = u_2(\mathbf{p}, e^2)$$

Applying the support theorem to Bach-or-Stravinsky (BoS)

		B	S
			
	B	2, 1	0, 0
	S	0, 0	1, 2

If there exists a Nash equilibrium with mixed strategies, in the form $((p_1, 1-p_1), (q_1, 1-q_1))$, with $p_1, q_1 \in (0, 1)$, it should hold that

- $2q_1 = 1 - q_1 \Rightarrow q_1 = 1/3$
- $p_1 = 2(1 - p_1) \Rightarrow p_1 = 2/3$
- The conditions for pl. 1 give us the mixed strategy of pl. 2
- Similarly the conditions for pl. 2 give the strategy of pl. 1
- Hence we have the mixed equilibrium $((2/3, 1/3), (1/3, 2/3))$

From 2x2 to 2xn games

	t_1	t_2	t_3	t_4
s_1	3, -2	1, 2	4, 6	2, 8
s_2	1, 12	5, 10	2, 4	3, -4

- What are the Nash equilibria in this game?
- There is no Nash equilibrium with pure strategies, hence, there must be one with mixed strategies
- We will start with pl. 1
 - i.e., with the player who has 2 pure strategies
- We are looking for a strategy $\mathbf{p} = (p_1, p_2) = (p_1, 1 - p_1)$ of pl. 1

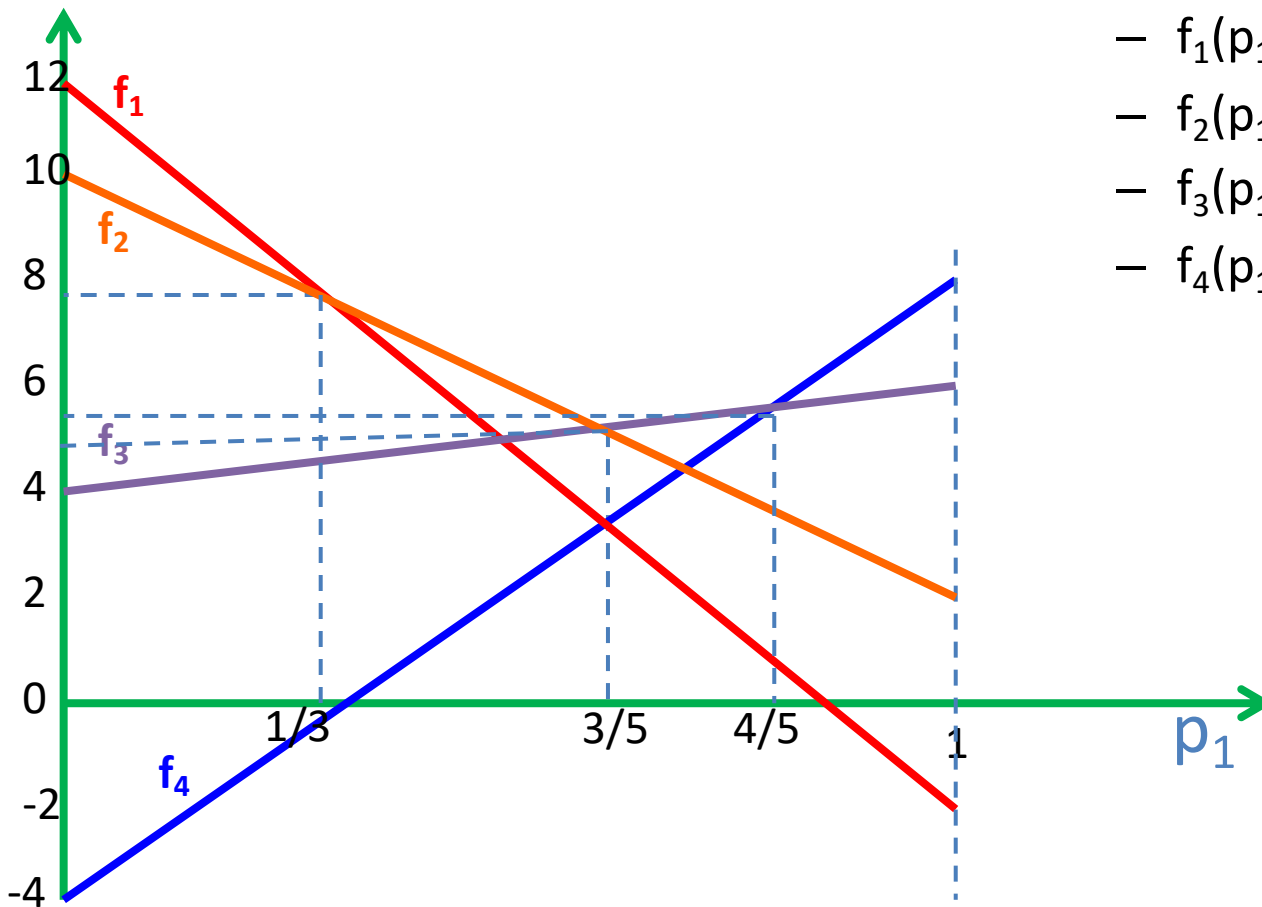
Analysis of 2xn games

	t_1	t_2	t_3	t_4
S_1	3, -2	1, 2	4, 6	2, 8
S_2	1, 12	5, 10	2, 4	3, -4

- Step 1: We look at pl. 2 and compute the terms
 - $u_2(\mathbf{p}, e^1) = f_1(p_1) = -14p_1 + 12,$
 - $u_2(\mathbf{p}, e^2) = f_2(p_1) = -8p_1 + 10,$
 - $u_2(\mathbf{p}, e^3) = f_3(p_1) = 2p_1 + 4$
 - $u_2(\mathbf{p}, e^4) = f_4(p_1) = 12p_1 - 4$

Analysis of 2xn games

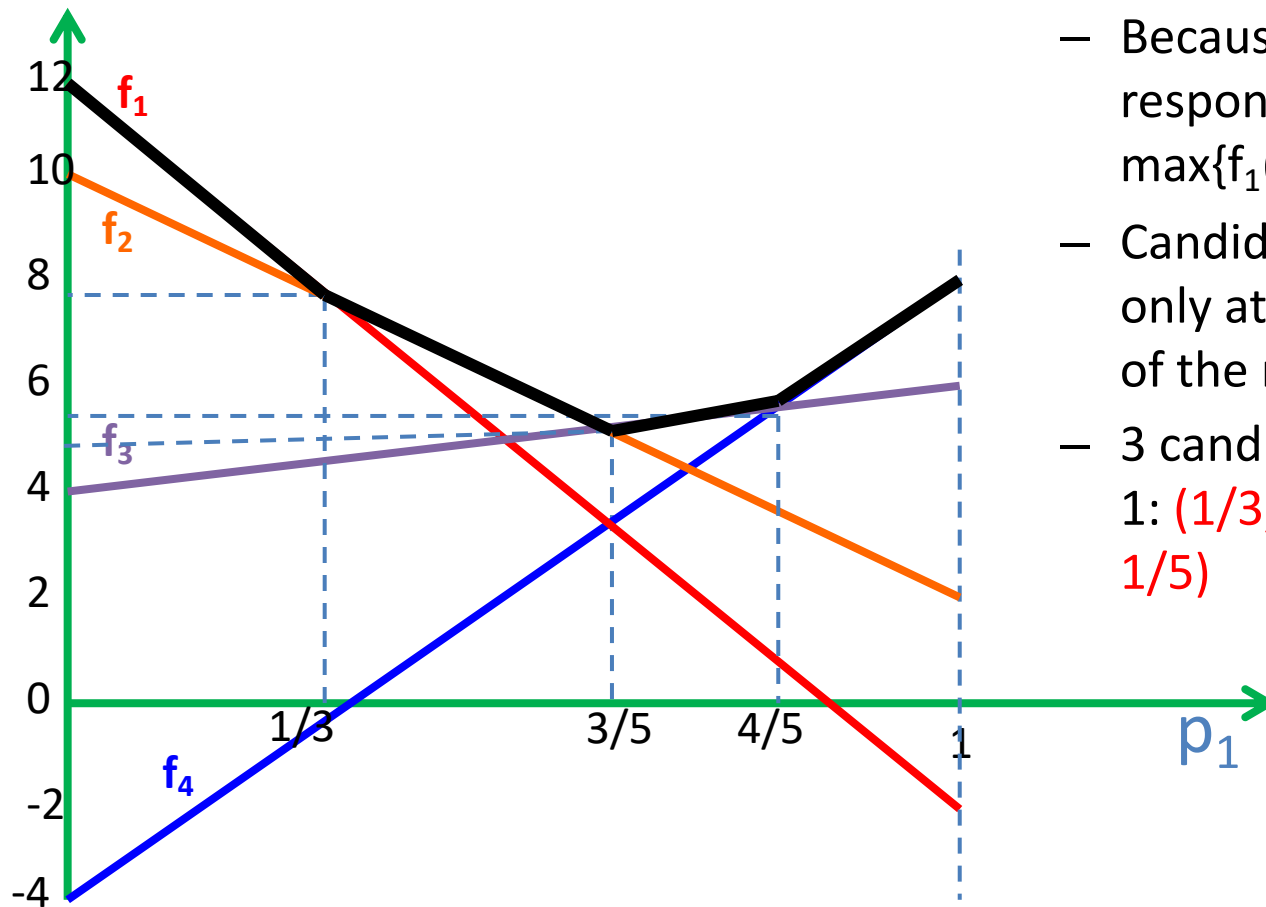
Step 2: Graphical representation



- $f_1(p_1) = -14p_1 + 12,$
- $f_2(p_1) = -8p_1 + 10,$
- $f_3(p_1) = 2p_1 + 4$
- $f_4(p_1) = 12p_1 - 4$

Analysis of 2xn games

Step 3: Candidate strategies for pl. 1



- Because pl. 2 will play a best response, we look at $\max\{f_1(p_1), f_2(p_1), f_3(p_1), f_4(p_1)\}$
- Candidate strategies for pl. 1 only at the intersection points of the max function
- 3 candidate strategies for pl. 1: $(1/3, 2/3)$, $(3/5, 2/5)$, $(4/5, 1/5)$

Analysis of 2xn games

	t_1	t_2	t_3	t_4
s_1	3, -2	1, 2	4, 6	2, 8
s_2	1, 12	5, 10	2, 4	3, -4

- Step 4: We check all the candidate strategies to see if they can yield an equilibrium

1st candidate strategy of pl. 1: $(1/3, 2/3)$

- We will search for a strategy of pl. 2 in the form: $\mathbf{q} = (q_1, 1 - q_1, 0, 0)$
- Since from the diagram, the 1st and 2nd columns are the best responses of pl. 2 to the strategy of pl. 1
- From the support theorem, it must hold that $u_1(e^1, \mathbf{q}) = u_1(e^2, \mathbf{q})$
- $3q_1 + 1 - q_1 = q_1 + 5(1 - q_1) \Rightarrow q_1 = 2/3$
- Since we found a valid probability, we have found a Nash equilibrium

Analysis of 2xn games

	t_1	t_2	t_3	t_4
s_1	3, -2	1, 2	4, 6	2, 8
s_2	1, 12	5, 10	2, 4	3, -4

- Step 4: We check all the candidate strategies to see if they can yield an equilibrium

2nd candidate strategy of pl. 1: $(3/5, 2/5)$

- We will search for a strategy of pl. 2 in the form: $\mathbf{q} = (0, q_2, 1 - q_2, 0)$
- Since from the diagram, the 2nd and 3rd columns are the best responses against the strategy of pl. 1
- From the support theorem, it should hold that $u_1(e^2, \mathbf{q}) = u_1(e^3, \mathbf{q})$
- By solving this, we get $q_2 = 1/3$
- Since we found a valid probability, we have found one more equilibrium

Analysis of 2xn games

	t_1	t_2	t_3	t_4
s_1	3, -2	1, 2	4, 6	2, 8
s_2	1, 12	5, 10	2, 4	3, -4

- Step 4: We check all the candidate strategies to see if they can yield an equilibrium

3rd candidate strategy of pl. 1: $(4/5, 1/5)$

- We will search for a strategy of pl. 2 of the form: $\mathbf{q} = (0, 0, q_3, 1 - q_3)$
- In a similar way, we get $q_3 = 1/3$
- Hence we have a 3rd Nash equilibrium

Analysis of 2xn games

	t_1	t_2	t_3	t_4
S_1	3, -2	1, 2	4, 6	2, 8
S_2	1, 12	5, 10	2, 4	3, -4

- In total: 3 Nash equilibria
 - $((1/3, 2/3), (2/3, 1/3, 0, 0))$
 - $((3/5, 2/5), (0, 1/3, 2/3, 0))$
 - $((4/5, 1/5), (0, 0, 1/3, 2/3,))$

A modified example

	t_1	t_2	t_3	t_4
S_1	3, -2	5, 2	4, 6	2, 8
S_2	1, 12	1, 10	2, 4	3, -4

- Suppose we change some of the payoffs of pl. 1 (here we changed the 2nd column)
- Which parts of the analysis change?
 - **Observation:** The candidate mixed strategies of pl. 1 were determined by the payoff matrix of pl. 2!
 - Hence, steps 1-3 remain exactly the same
 - Again, 3 candidate strategies for pl. 1

A modified example

	t_1	t_2	t_3	t_4
S_1	3, -2	5, 2	4, 6	2, 8
S_2	1, 12	1, 10	2, 4	3, -4

- Step 4: We check all the candidate strategies to see if they can yield an equilibrium

1st candidate strategy of pl. 1: $(1/3, 2/3)$

- We will search for a strategy of pl. 2 in the form: $\mathbf{q} = (q_1, 1 - q_1, 0, 0)$
- From the support theorem, it must hold that $u_1(e^1, \mathbf{q}) = u_1(e^2, \mathbf{q})$
- $3q_1 + 5(1 - q_1) = q_1 + 1 - q_1 \Rightarrow q_1 = 2$
- Not a valid probability!
- Hence, this candidate strategy does not yield an equilibrium

A modified example

	t_1	t_2	t_3	t_4
S_1	3, -2	5, 2	4, 6	2, 8
S_2	1, 12	1, 10	2, 4	3, -4

- Step 4: We check all the candidate strategies to see if they can yield an equilibrium

2nd candidate strategy of pl. 1: $(3/5, 2/5)$

- We will search for a strategy of pl. 2 in the form : $\mathbf{q} = (0, q_2, 1 - q_2, 0)$
- From the support theorem, it should hold that $u_1(e^2, \mathbf{q}) = u_1(e^3, \mathbf{q})$
- $5q_2 + 4(1 - q_2) = q_2 + 2(1 - q_2) \Rightarrow q_2 = -1$
- Not a valid probability
- Hence, no equilibrium

A modified example

	t_1	t_2	t_3	t_4
S_1	3, -2	5, 2	4, 6	2, 8
S_2	1, 12	1, 10	2, 4	3, -4

- Step 4: We check all the candidate strategies to see if they can yield an equilibrium

3rd candidate strategy of pl. 1: $(4/5, 1/5)$

- Since we have not found any other equilibrium, Nash's theorem guarantees that now we will find one
- We will search for a strategy of pl. 2 of the form: $\mathbf{q} = (0, 0, q_3, 1 - q_3)$
- In the modified example, columns 3 and 4 have not changed
- Hence, we will arrive at the same result: $q_3 = 1/3$
- Unique Nash equilibrium: $((4/5, 1/5), (0, 0, 1/3, 2/3,))$

Back to nxm games

- Summarizing known algorithms:
 - Brute-force, based on the support theorem, worst case: need to solve $O(2^{n+m})$ linear programs
 - [Lemke, Howson '64], worst case: still exponential
 - Other approaches: [Kuhn '61, Mangasarian '64, Lemke '65], also exponential worst case running time
- Polynomial time algorithms only for special cases
 - 0-sum games
 - 2xn games
 - Games with constant rank payoff matrices
- We are not aware of any polynomial time algorithm for general nxm normal form games!

Algorithms for normal form games

- Could it be that the problem is **NP**-complete?
- Probably not
 - [Megiddo, Papadimitriou '89]: strong evidence that it cannot be **NP**-complete
 - If it were \Rightarrow **NP** = co-**NP** (highly unlikely to be true)
- It is **NP**-complete if we add more requirements
 - E.g. Find a Nash equilibrium that maximizes the sum of the utilities
[Gilboa, Zemel '89, Conitzer, Sandholm '03]
 - A different problem than just finding a Nash equilibrium
- Further issues
 - There exist games, with integer payoff matrices, and with ≥ 3 players, where the probabilities in their Nash equilibria are irrational numbers
[Nash '51]
 - Hence, we cannot even represent the mixed strategies by a finite number of bits

Back to the proof of Nash's theorem

- Theorem [Nash 1951]: Every finite game possesses at least one equilibrium when we allow mixed strategies
- Nash's proof reduces to using Brouwer's fixed point theorem
- Brouwer's theorem reduces to using Sperner's lemma

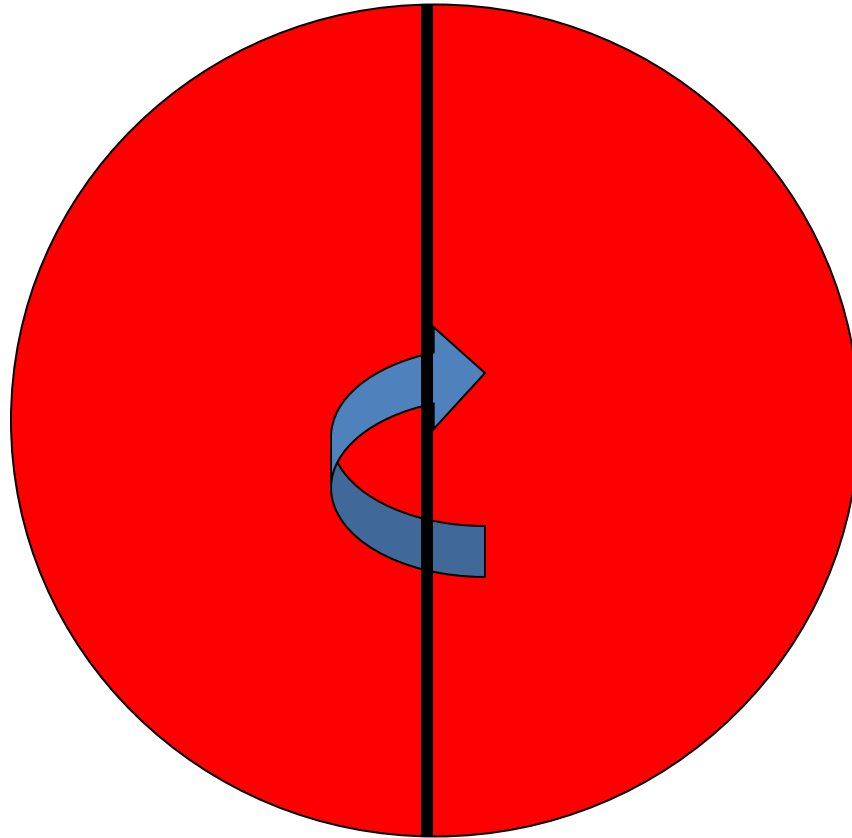
Brouwer's theorem

- **Brouwer's theorem:** Let $f:D \rightarrow D$, be a continuous function, and suppose D is convex and compact. Then there exists x such that $f(x) = x$

Illustrations of Brouwer's theorem

Suppose D is a disc

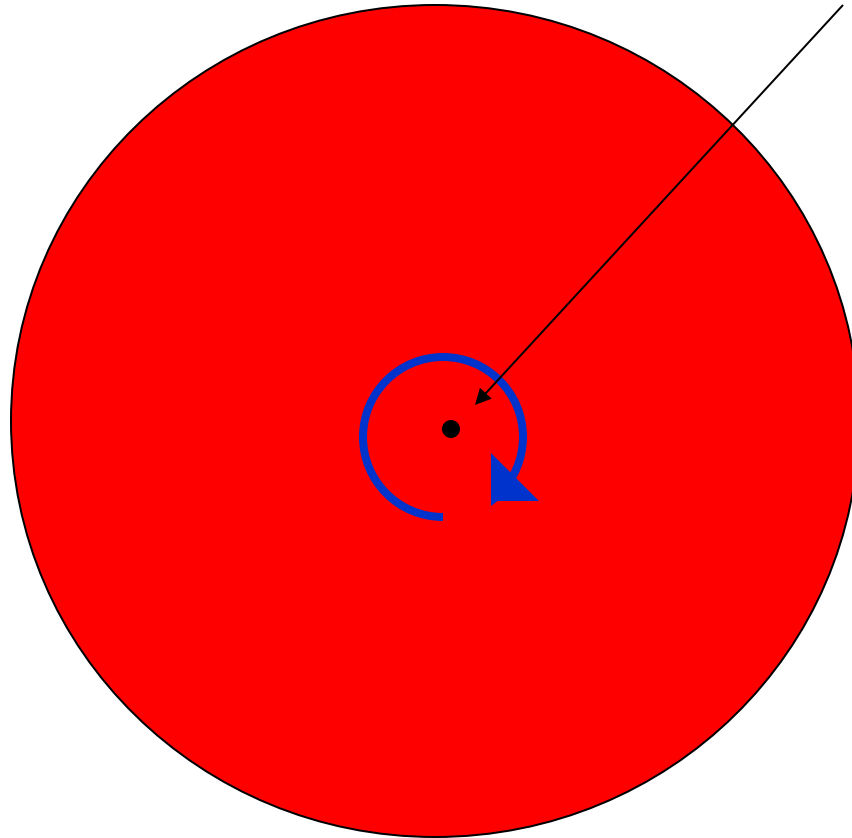
Flip?



Illustrations of Brouwer's theorem

Suppose D is a disc

Rotate?



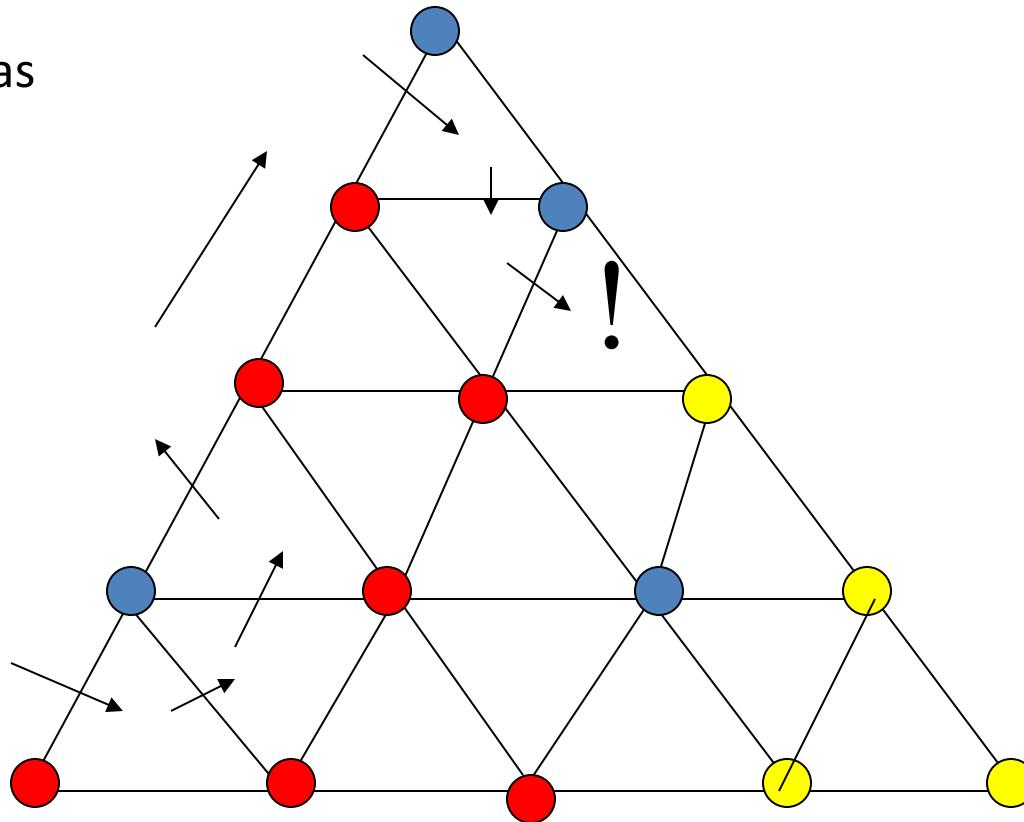
Sperner's lemma

In 2 dimensions

- Let D be the 2-dimensional simplex
 - $D = \{(x_1, x_2, x_3) : x_1 + x_2 + x_3 = 1, x_i \geq 0, \text{ for } i=1, 2, 3\}$
 - D is a triangle
- Consider a triangulation of D
- Color all the vertices of the small triangles, using 3 colors such that:
 - The 3 vertices of D have a different color
 - Along each edge of D , we use only the colors of the 2 vertices of the edge (1 color forbidden)
 - No restriction for the interior of D

Sperner's lemma

**Sperner's
Lemma:** Any
such coloring has
at least one
trichromatic
triangle



Algorithms for normal form games

Let us look at the computational problems:

- **SPERNER:** Given a coloring satisfying the conditions of Sperner's lemma, find a trichromatic triangle
- **BROUWER:** Given a function satisfying the conditions of Brouwer's theorem, find a fixed point
- **NASH:** Given a finite normal form game, find a Nash equilibrium

What is common with all 3?

- They are search problems, where we know a solution always exists

Complexity classes for search problems

Informal descriptions

- **FP (Function P)**: The version of P for search problems
- **FNP (Function NP)**: The version of NP for search problems
- **TFNP (Total FNP)**: The class of search problems that always have a solution

Fact: $FP \subseteq TFNP \subseteq FNP$

Complexity classes for search problems

- TFNP has several interesting subclasses
- Depending on how the proof of existence is established
- PLS (Polynomial time Local Search)
- PPA (Polynomial time Parity Argument)
- PPAD (Polynomial time Parity Argument, Directed)
- PPP (Polynomial time Pigeonhole Principle)
- And more...

In fact, our problems belong to one of these subclasses

The class PPAD

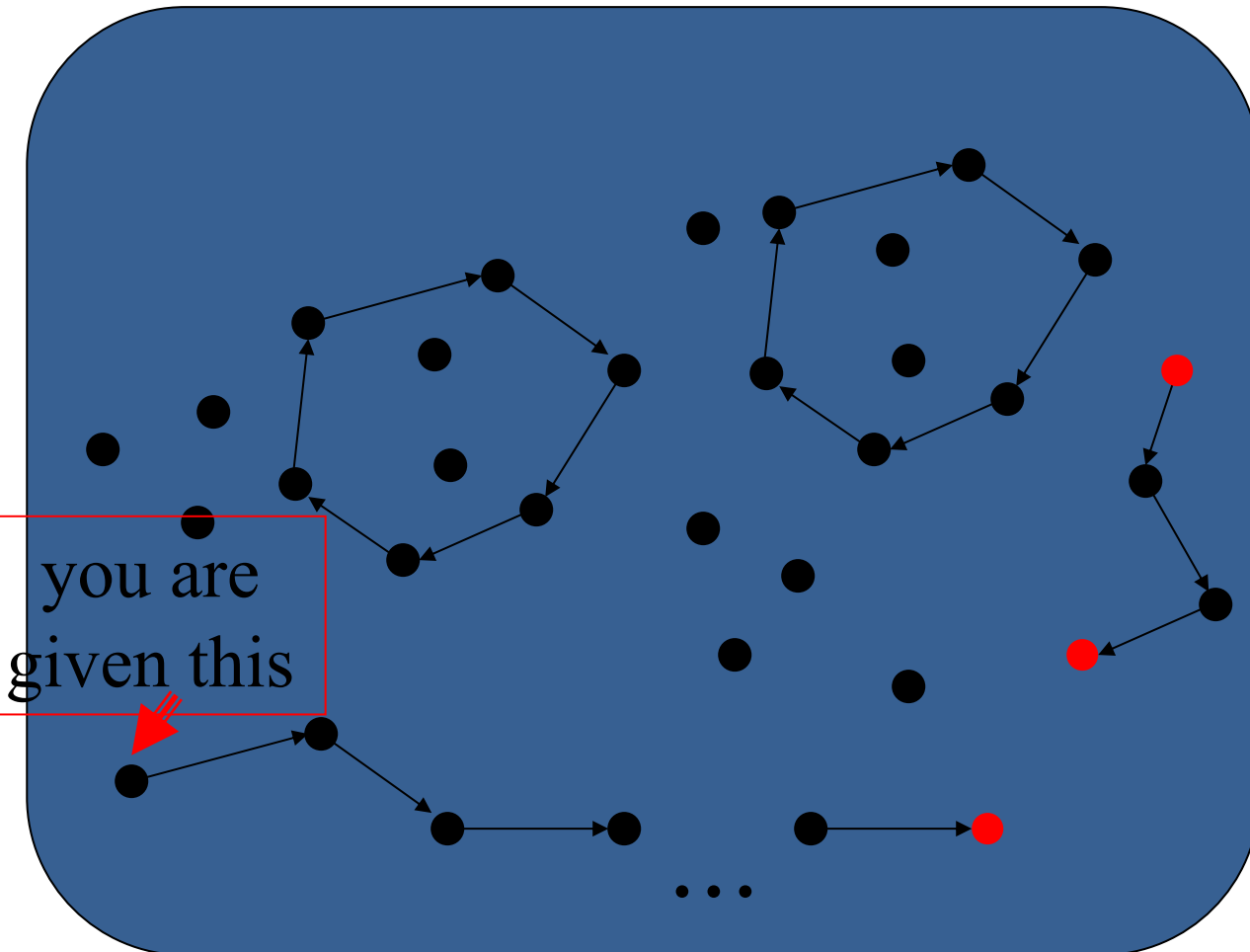
[Papadimitriou '94]

- Consists of problems where the existence of a solution can be established by a particular kind of parity argument
- Namely, PPAD contains all problems that can be reduced to:

END OF THE LINE:

- We are given a directed graph with $\text{in-degree}(u), \text{out-degree}(u) \leq 1$ for every vertex u
- The graph is given implicitly by two circuits P, C
 - (u,v) is an edge iff $u = P(v)$ and $v = C(u)$
 - i.e., we are **only** allowed to ask **queries** for the successor or the predecessor of a node (at most polynomially many queries)
- We are also given a source node ($\text{in-degree}=0$)
- *Goal: Find the sink, or another source*
 - existence of such a node is guaranteed, by a parity argument: **the total number of sources and sinks is even**

The class PPAD



Q: Is there an efficient algorithm for finding another unbalanced node **without actually following the path?**

Complexity of finding a Nash equilibrium

- Open problem for many years
- Eventually:
 - The problem belongs to **PPAD**
 - Membership in PPAD is established via the Lemke-Howson algorithm
 - [Daskalakis, Goldberg, Papadimitriou, September 2005]: **PPAD**-complete for 4-player games, conjectured that for 2 players there is an efficient algorithm
 - [Chen, Deng, November 2005]: **PPAD**-complete even for 2-player games!
 - [Chen, Deng, Teng, February 2006]: **PPAD**-complete even for some approximate versions of equilibria
 - Current belief is that problems in **PPAD** are not poly-time solvable
 - Finding an exact Nash equilibrium is most probably intractable

Other PPAD-complete problems

How can we define **BROUWER** as a computational problem?

- Consider a function f that satisfies the conditions of Brouwer's theorem
 - It may not be easy to succinctly describe f as input to the algorithm
 - Also, the fixed point may contain irrational numbers
- Thus, the function is given implicitly via a circuit (only allowed to ask queries for the value of the function at any point of the domain)
- **Goal:** Find an approximate fixed point: a point x such that $|f(x) - x| < \epsilon$

Theorem: BROUWER is PPAD-complete

Finding a Nash equilibrium is **equivalent** to finding approximate fixed points of continuous functions

- Note that the proof of Nash's theorem only showed that finding an equilibrium is at most as difficult as finding fixed points

Approximate Nash equilibria

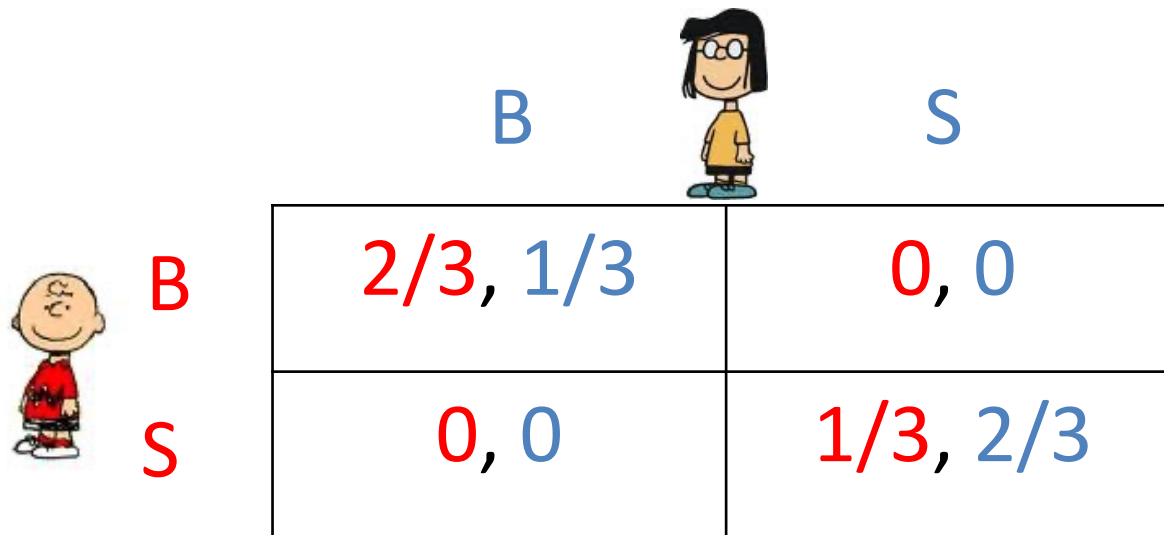
Approximate Nash equilibria



- Since the problem of computing equilibria is hard, we can consider possible relaxations of the initial definition
- Recall the definition of Nash equilibria: A profile of mixed strategies (\mathbf{p}, \mathbf{q}) is a Nash equilibrium if
 - $u_1(\mathbf{p}, \mathbf{q}) \geq u_1(\mathbf{e}^i, \mathbf{q})$ for every pure strategy \mathbf{e}^i of pl. 1
 - $u_2(\mathbf{p}, \mathbf{q}) \geq u_2(\mathbf{p}, \mathbf{e}^j)$ for every pure strategy \mathbf{e}^j of pl. 2

Approximate Nash equilibria

- Definition: A profile of mixed strategies (\mathbf{p}, \mathbf{q}) is an ε -Nash equilibrium if
 - $u_1(\mathbf{p}, \mathbf{q}) \geq u_1(\mathbf{e}^i, \mathbf{q}) - \varepsilon$, for every pure strategy \mathbf{e}^i of pl. 1
 - $u_2(\mathbf{p}, \mathbf{q}) \geq u_2(\mathbf{p}, \mathbf{e}^j) - \varepsilon$, for every pure strategy \mathbf{e}^j of pl. 2
- **In words**: a profile of strategies is an ε -Nash equilibrium if no player can gain more than ε by deviating
- When we study ε -Nash equilibria, we usually normalize the utilities to be in $[0, 1]$
 - Thus also $\varepsilon \in [0, 1]$

Example of approximate Nash equilibria



		 B S	
	B	2/3, 1/3	0, 0
	S	0, 0	1/3, 2/3

Consider the profile $(\mathbf{p}, \mathbf{q}) = ((0.6, 0.4), (0.4, 0.6))$

- $u_1(\mathbf{p}, \mathbf{q}) = 0.6 \times 0.4 \times 2/3 + 0.4 \times 0.6 \times 1/3 = 0.24$
- $u_1(\mathbf{e}^1, \mathbf{q}) = 0.4 \times 2/3 = 0.267 = u_1(\mathbf{p}, \mathbf{q}) + 0.027$
- $u_1(\mathbf{e}^2, \mathbf{q}) = 0.6 \times 1/3 = 0.2 < 0.24$
- Similar analysis for pl. 2
- Hence, this profile is a 0.027-Nash equilibrium

None of the players can gain more than 0.027 by deviating to another strategy

Approximate Nash equilibria

A stronger notion of approximation

- In words: a profile of strategies (\mathbf{p}, \mathbf{q}) is an ε -well-supported Nash equilibrium if any strategy from $\text{Supp}(\mathbf{p})$ is an approximate best response to \mathbf{q} and vice versa
- Formally: (\mathbf{p}, \mathbf{q}) is an ε -well-supported Nash equilibrium if:
 - $u_1(\mathbf{e}^i, \mathbf{q}) \geq u_1(\mathbf{e}^k, \mathbf{q}) - \varepsilon$, for every $i \in \text{Supp}(\mathbf{p})$ and every $k \in \{1, 2, \dots, n\}$
 - $u_2(\mathbf{p}, \mathbf{e}^j) \geq u_2(\mathbf{p}, \mathbf{e}^k) - \varepsilon$, for every $j \in \text{Supp}(\mathbf{q})$ and every $k \in \{1, 2, \dots, n\}$

Fact: An ε -well-supported Nash equilibrium is also an ε -Nash equilibrium (but not vice versa)

Searching for Approximate Equilibria

- We will focus on the simpler version of ϵ -Nash equilibria
- At the same time, we also want to focus on strategy profiles that are simple, and easy to describe

Definition: A k -uniform strategy is a strategy where all probabilities are integer multiples of $1/k$

e.g. $(3/k, 0, 0, 1/k, 5/k, 0, \dots, 6/k)$

Important observation: Support size of a k -uniform strategy $\leq k$

Can we have approximate equilibria with k -uniform strategies for small values of k ?

A Subexponential Algorithm (Quasi-PTAS)

Theorem [Lipton, Markakis, Mehta '03]: Consider a $n \times n$ game. For any ε in $(0,1)$, and for every $k \geq 9 \log n / \varepsilon^2$, there exists a pair of k -uniform strategies (\mathbf{p}, \mathbf{q}) that forms an ε -Nash equilibrium

Lesson learnt: there is no need to use a big support!

- For 0-sum games already proved in [Althofer '94, Lipton, Young '94]

Proof idea:

- Use of the "Probabilistic Method"
- Sample a mixed strategy for each player according to the distribution of a Nash equilibrium
 - Feasible because of Nash's theorem
- Then prove that with positive probability the desired property holds

A Subexponential Algorithm (Quasi-PTAS)

Theorem [Lipton, Markakis, Mehta '03]: Consider a $n \times n$ game. For any ε in $(0,1)$, and for every $k \geq 9 \log n / \varepsilon^2$, there exists a pair of k -uniform strategies (\mathbf{p}, \mathbf{q}) that forms an ε -Nash equilibrium

Corollary : We can compute an ε -Nash equilibrium in time

$$n^{O(\log n / \varepsilon^2)}$$

Proof of Corollary: There are $n^{O(k)}$ pairs of supports to look at. Verify the ε -equilibrium condition.

Generalizations

- The same property holds for ε -well-supported equilibria as well [Kontogiannis, Spirakis '10]
- For m -player games with n pure strategies per player, the same technique yields an algorithm for approximate Nash equilibria with
 - support size: $k = O(m^2 \log(m^2 n)/\varepsilon^2)$
 - running time: exponential in $\log n, m, 1/\varepsilon$
- Previously known approximations:
 - [Scarf '67]: exponential in $n, m, \log(1/\varepsilon)$ (via fixed point approximations)

An application

[McCarthy, Laan, Wang, Vayanos, Sinha, Tambe '18]

- **Threat Screening Games:** Games for modeling decision problems related to screening at airports, borders, and other areas
- Motivated by a collaboration with the US Transportation Security Administration
- Use of mixed strategies for selecting how to screen quite popular during last years
- **Main practical result:** Simulations for screening in a large airport (comparable to the Los Angeles International Airport) show that approximate equilibria with k -uniform small support strategies behave very well and have the potential to be deployed in practice

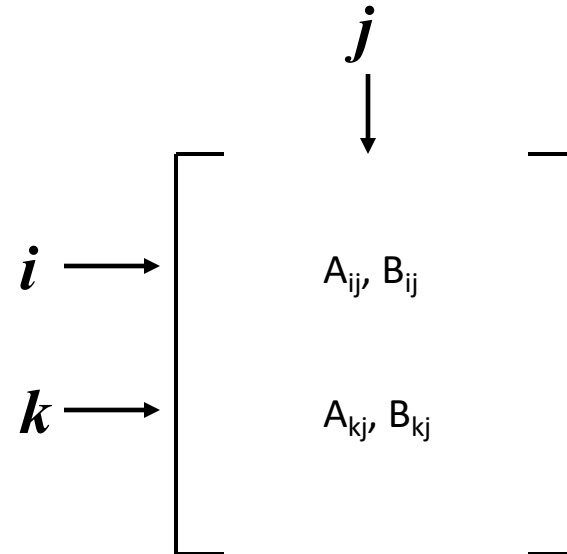
Moving on...

- How good is an algorithm with running time $n^{O(\log n / \epsilon^2)}$?
- For sure better than exponential
 - Better than n^n or 2^n
- But still not polynomial running time
 - Usually referred to as quasi-polynomial
- For what values of ϵ can we have polynomial time algorithms?

Polynomial Time Approximation Algorithms

For $\varepsilon = 1/2$:

- Pick arbitrary row i
- Let $j = \text{BR}(i)$ = best response to i
- Find $k = \text{BR}(j)$, pl. 1 plays i or k with prob. $1/2$ each
- Pl. 2 just plays j



Proposition: This is a $1/2$ -approximate equilibrium with support size ≤ 2 for both players!

[Feder, Nazerzadeh, Saberi '07]: For $\varepsilon < 1/2$, we need in worst case, support at least $\Omega(\log n)$

Polynomial Time Approximation Algorithms

Better than $\frac{1}{2}$ -approximations in polynomial time

[Daskalakis, Mehta, Papadimitriou '07]: polynomial time algorithm for $\varepsilon = 1 - 1/\varphi = (3 - \sqrt{5})/2 \approx 0.382$ (φ = golden ratio)

- Based on sampling + Linear Programming
- Need to solve polynomial number of linear programs
- Not a very fast algorithm
- Polynomial time algorithm but still a large number of linear programs to be solved

Polynomial Time Approximation Algorithms

[Bosse, Byrka, Markakis '07]: a different LP-based method with the same approximation of 0.382

- Needs to solve only 1 linear program
- Similar idea in [Kontogiannis, Spirakis '07] for well-supported approximation
- A small tweak can also yield a better approximation of 0.36

Recall: 0-sum games can be solved in polynomial time (equivalent to linear programming)



- Given a game defined by the arrays (A, B) , start with an equilibrium of the 0-sum game $(A-B, B-A)$

- If incentives to deviate are “high”, players adjust their strategies via best response moves

A 0.382-approximation algorithm

Parameters of the algorithm: $\alpha, \delta_2 \in [0,1]$

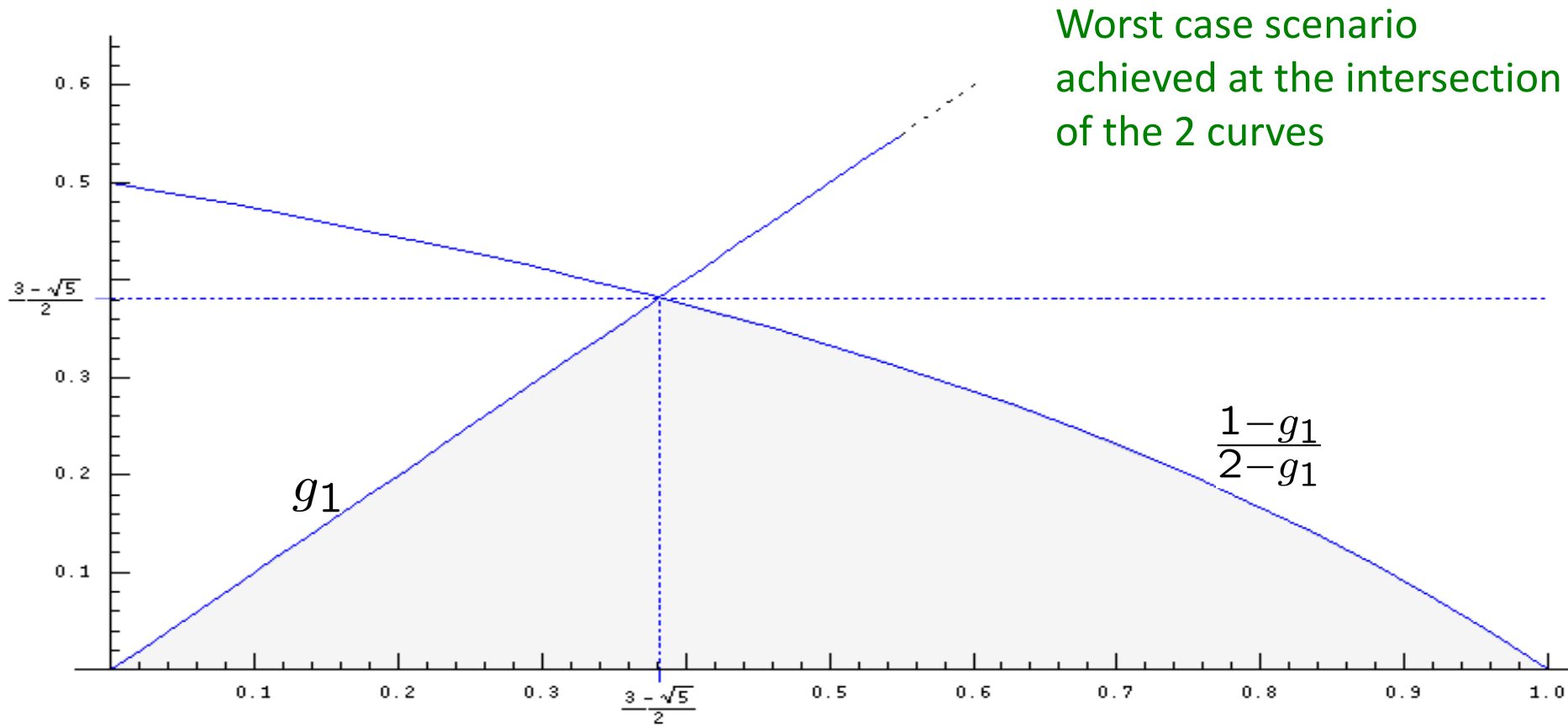
1. Find an equilibrium x^*, y^* of the 0-sum game $(A - B, B - A)$
2. Let g_1, g_2 be the maximum gain by deviating to a pure strategy for row and column player. Suppose $g_1 \geq g_2$
3. If $g_1 \leq \alpha$, output x^*, y^*
4. Else: let $b_1 = \text{best response to } y^*$, $b_2 = \text{best response to } b_1$
5. Output:

$$x = b_1$$

$$y = (1 - \delta_2) y^* + \delta_2 b_2$$

Theorem: The algorithm with $\alpha = 1 - 1/\varphi$ and $\delta_2 = (1 - g_1) / (2 - g_1)$ achieves a $(1 - 1/\varphi)$ -approximation

A 0.382-approximation algorithm



Yet another approach

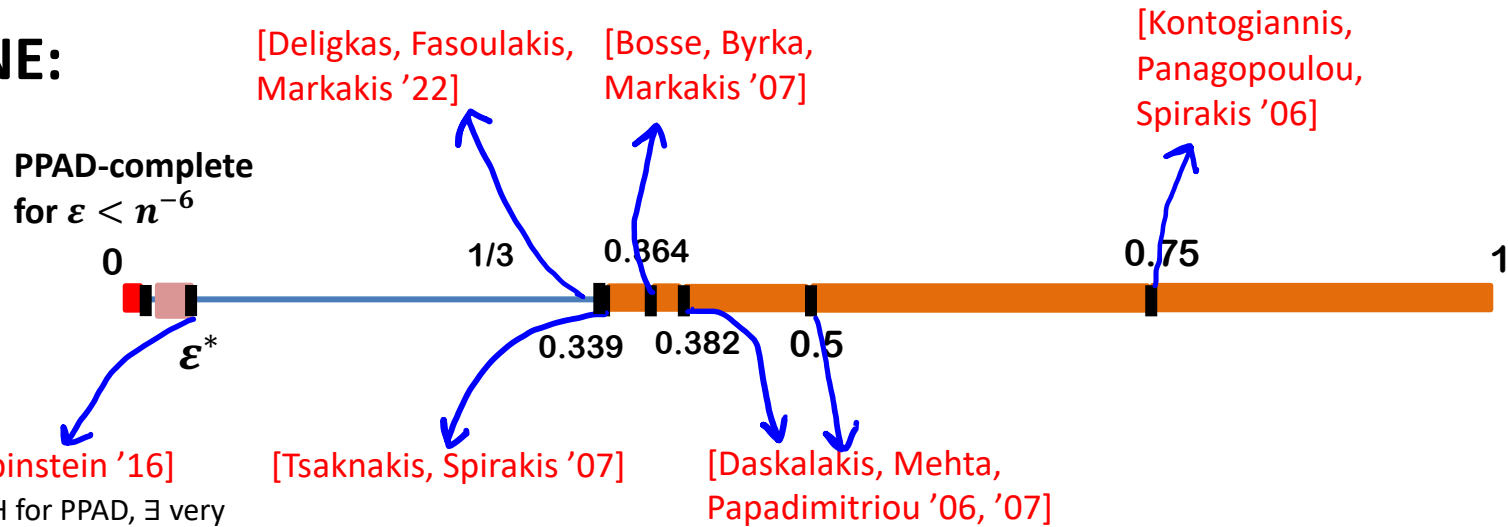
- **[Spirakis, Tsaknakis '07]:** algorithm with an approximation of $\epsilon = 0.339$
 - Best known approximation for many years, till recently
 - A different optimization approach, but yet another LP-based method
 - It starts with a descent-based method to identify a stationary point
 - Needs to solve a polynomial number of linear programs (one per each iteration)
- **[Deligkas, Fasoulakis, Markakis '22]:** Currently best approximation of $\epsilon = 1/3$
 - Based on a tweak of the Spirakis-Tsaknakis algorithm
 - Improving the bottleneck case of their algorithm
- **Big open problem:**
 - Can we find algorithms for lower values of ϵ , closer to 0?
 - Is it possible to have a poly-time algorithm for **any** constant $\epsilon > 0$?
 - **Probably not... [Rubinstein '16]**
- So far, there have been further improvements for several special classes of games
 - Low-rank matrices, sparse matrices, symmetric games, win-lose games, ...

Progress on other notions of approximation

- ϵ -well-supported equilibria:
 - [Kontogiannis, Spirakis '10]: Polynomial time only for $\epsilon = 2/3$, based also on solving 0-sum games
 - More recently improved to 0.6528 [Czumaj et al. '18]
 - And even more recently improved to $1/2$ [Deligkas, Fasoulakis, Markakis '23]
- Even stronger notion of approximation: require that the profile found is geometrically close to an exact Nash equilibrium
 - [Etessami, Yannakakis '07]: mostly negative results
- Open problem to provide more positive results, even for special cases, for these concepts as well

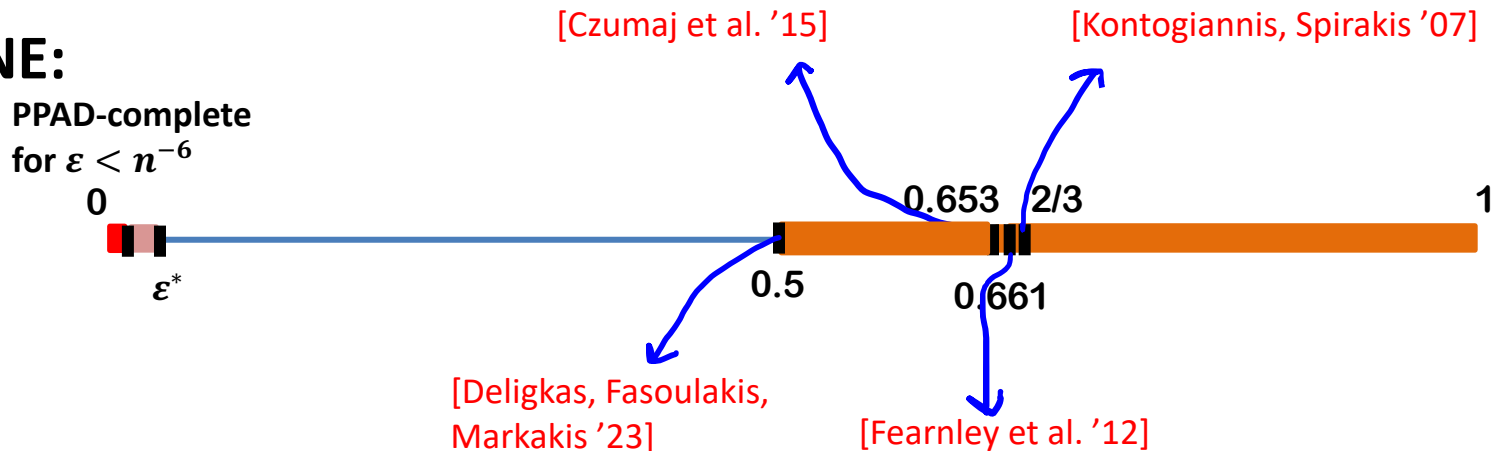
The story so far for ϵ -NE and ϵ -WSNE

ϵ -NE:



Under ETH for PPAD, \exists very small constant ϵ^* s.t. no poly-time algorithm for $\epsilon < \epsilon^*$

ϵ -WSNE:



Post-Mortem

- Difficult to find exact Nash equilibria for an arbitrary 2-player game
- A bit less difficult to find approximate Nash equilibria
 - But still challenging and not yet well understood
- Is it a catastrophe if we do not have efficient algorithms for every game?
 - Players in practice may also be able to adjust their strategies and gradually converge to an equilibrium by observing each other's actions
 - Still, “if your laptop cannot find an equilibrium, then neither can the market”, quote from Kamal Jain (2003)
- Despite the high complexity, the notion of a Nash equilibrium remains among the most important notions in game theory

Post-Mortem

- **Take-home story:** Nash equilibria form a good starting point from a conceptual point of view
- But when intractable, we should think towards alternative and tractable variations of equilibrium concepts
- Ongoing research:
 - Learning algorithms with convergence guarantees
 - Also connected to training neural networks
 - Many positive results for 0-sum games (starting with fictitious play [Robinson '51])
 - Not as easy for general games
 - No-regret algorithms provide convergence “on average”
 - Several variations of gradient descent under consideration during last 5 years...