Algorithmic Game Theory
CoReLab (NTUA)

Lecture 3:

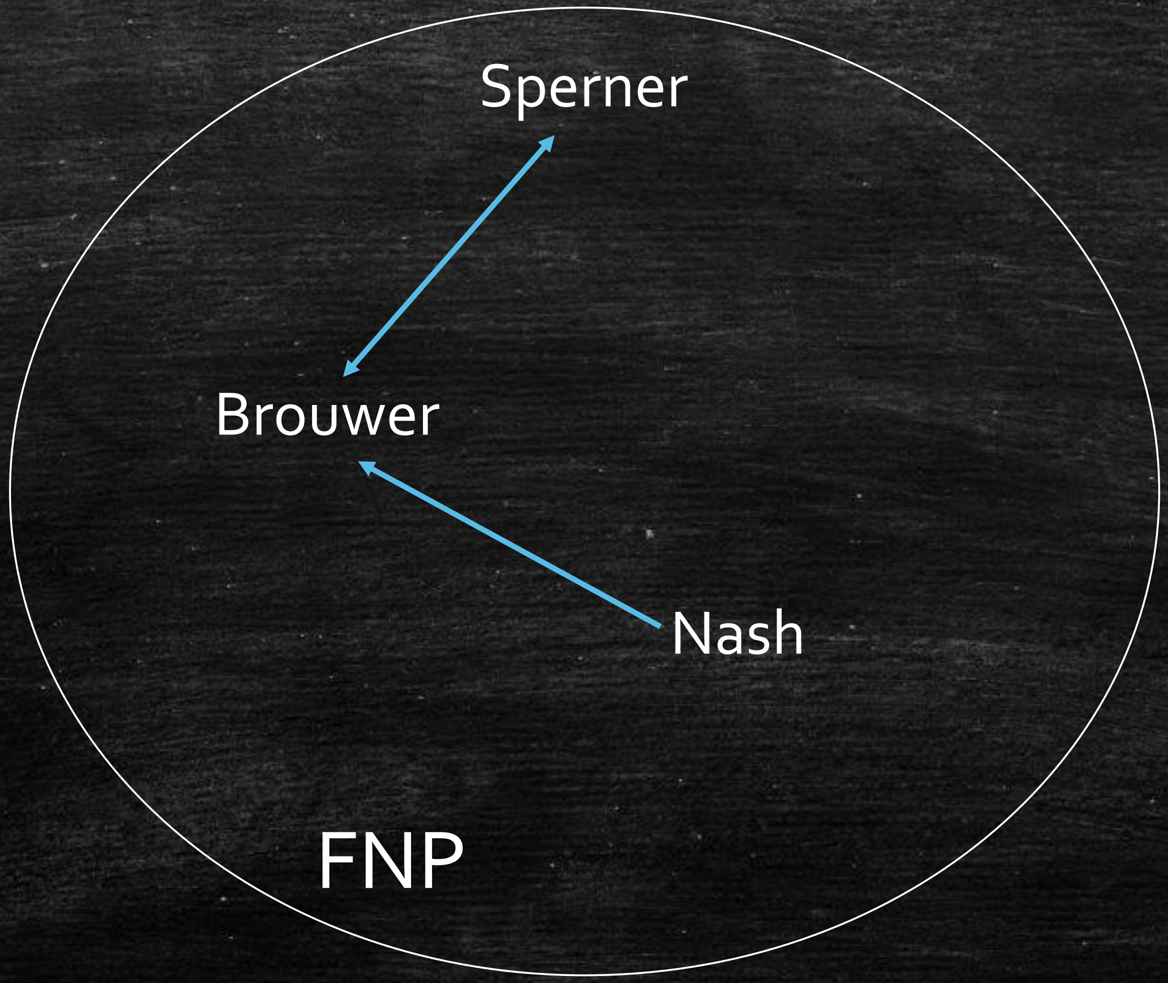
Tractability of Nash Equilibria
PPAD completeness
Lemke-Howson algorithm
LMM

So far

NE in 2-player zero sum	↔	LP Duality

Nash's Theorem (1950)	Every (finite) game has a Nash Equilibrium.
⬆	
Brouwer's Theorem (1911)	Every continuous function from a closed compact convex set to itself has a fixed point.
⬆	
Sperner's Lemma (1950)	Every proper coloring of a triangulation has a panchromatic triangle.
⬆	
Parity Argument (1990)	If a directed graph has an unbalanced node, then it must have another.

What we know



General 2-player games

A slightly more ambitious attempt would be to face general 2- player games and provide efficient algorithms or prove hardness results.

An other direction would be to face 3-player zero sum games….

but in fact these games can only be harder.(!)

Nash vs NP

The problem resisted polynomial algorithms for a long time which altered the research direction towards hardness results.

The first idea would be to prove Nash an FNP-complete problem.

But accepting an FSAT $\longrightarrow$ Nash

reduction directly implies NP=coNP. (!)

Nash vs TFNP

What prevented our previous attempt was the fact that Nash problem always has solution.

So the next idea would be to prove it complete for this class.

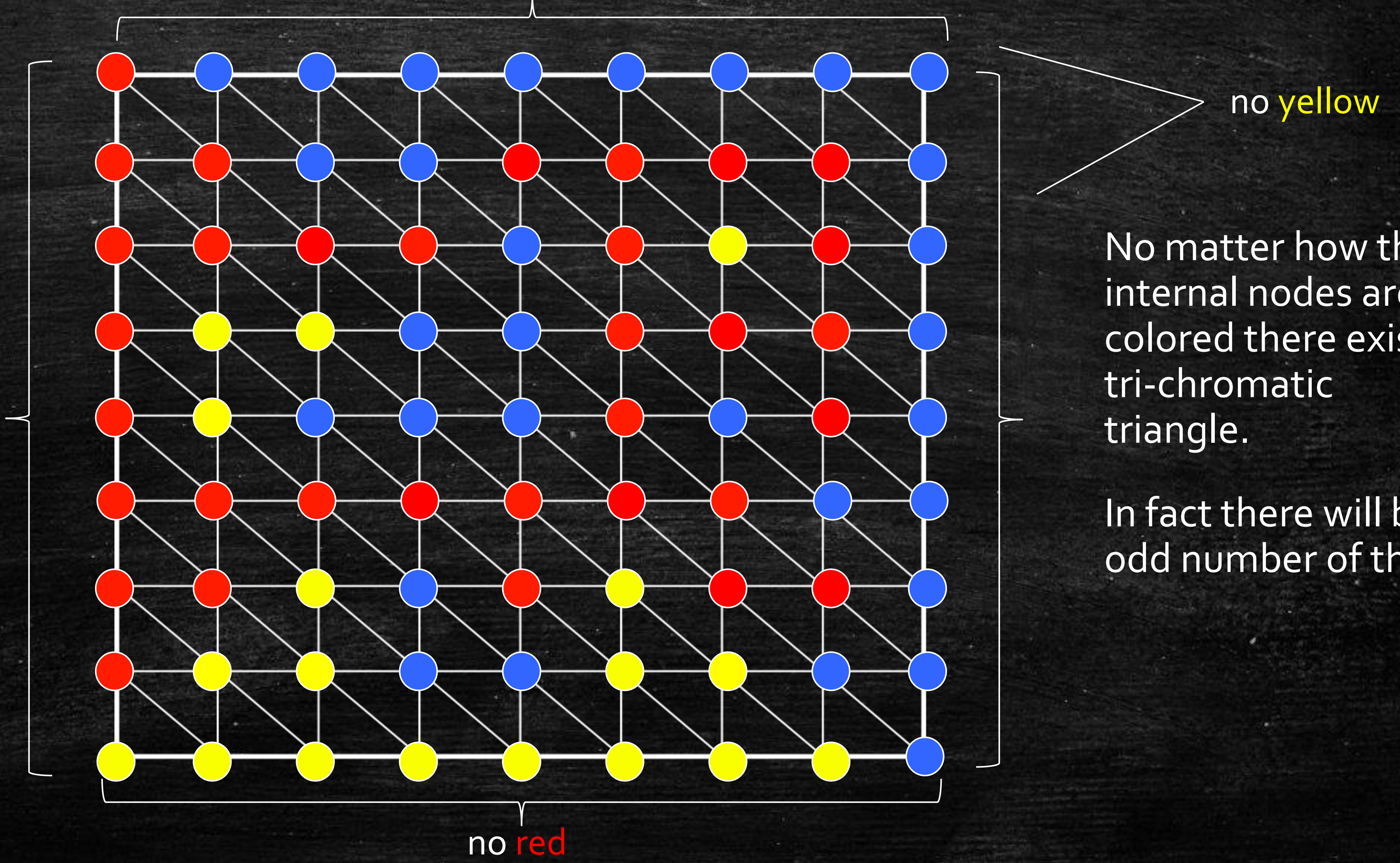
But no complete problem is known for TFNP.

Complexity Theory of Total Search Problems

In order to overcome the obstacles we face we need to work as follows:

1. Identify the combinatorial structure that makes our problems total.
2. Define a new complexity class inspired from our observation.
3. Check the 'tightness' of our class – in other words that our problems are complete for the class.

Sperner's Lemma revisited



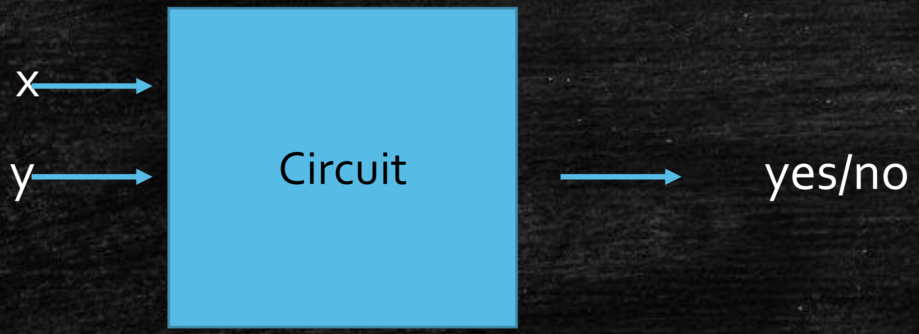
No matter how the internal nodes are colored there exists a tri-chromatic triangle.

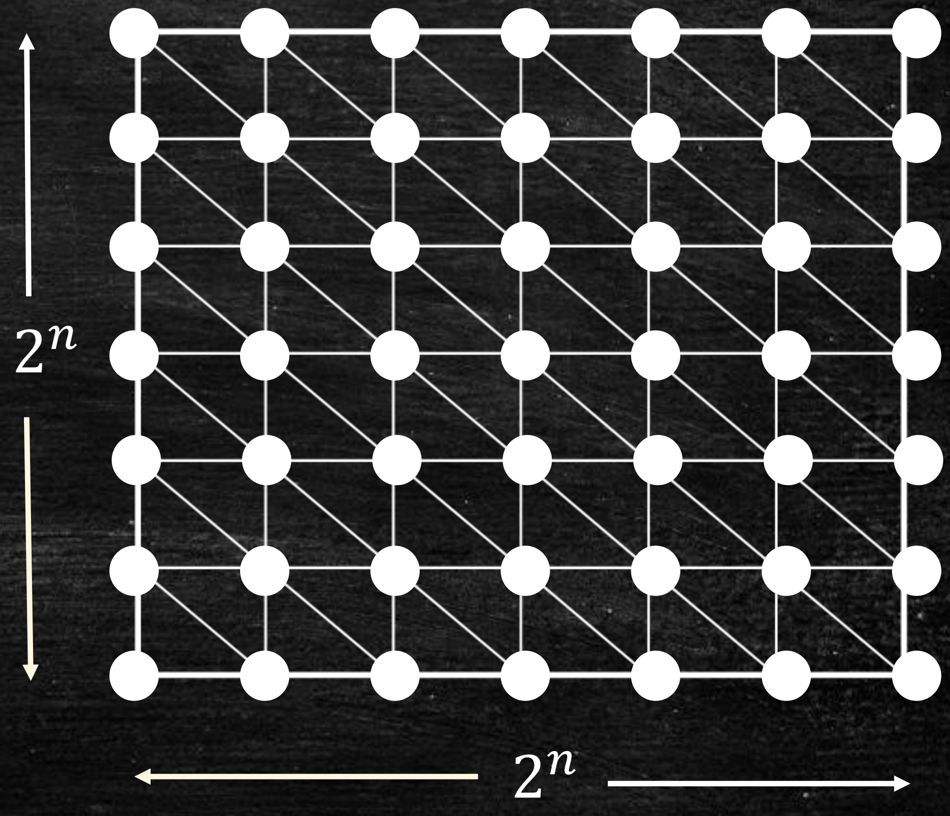
In fact there will be an odd number of them.

Why Sperner is hard?

We have to work with a graph of exponential size!

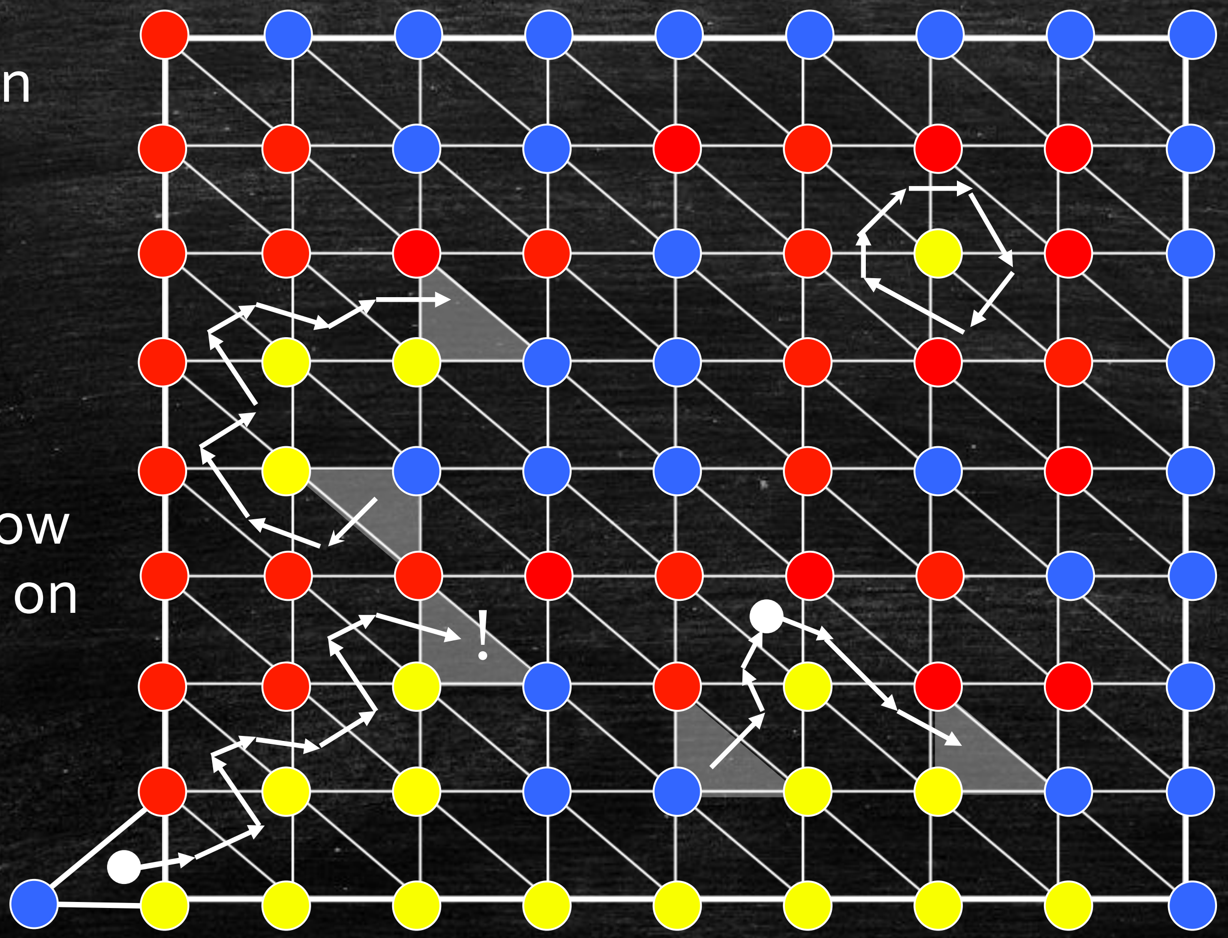
Input: 2 n-bit numbers





Proof of Sperner’s Lemma

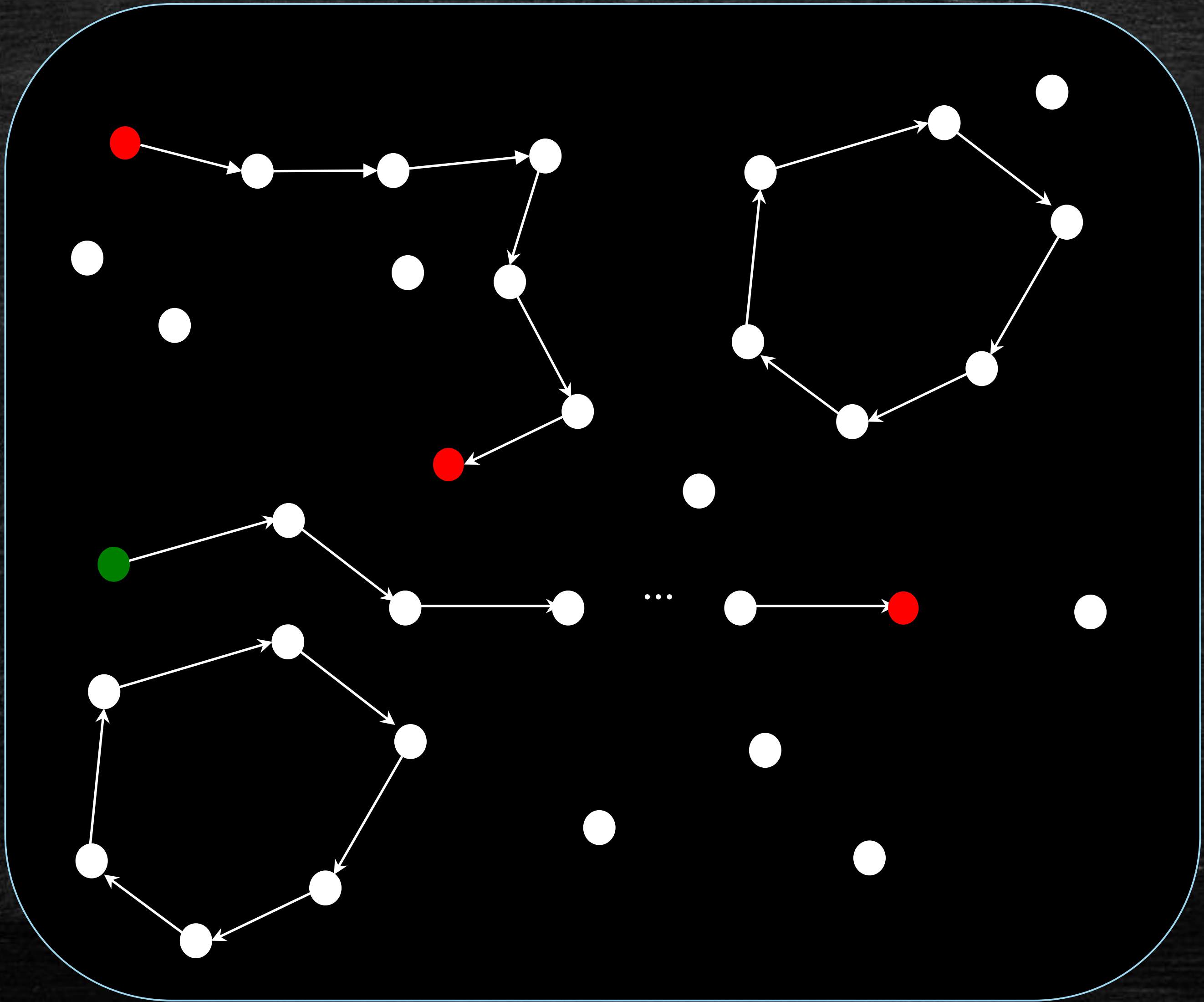
1. We introduce an artificial vertex on the bottom left
2. We define a directed walk crossing red-yellow doors having red on our left



- Claim: The walk can’t get out nor can it loop into itself
- It follows that there is an odd number of tri-chromatic triangles

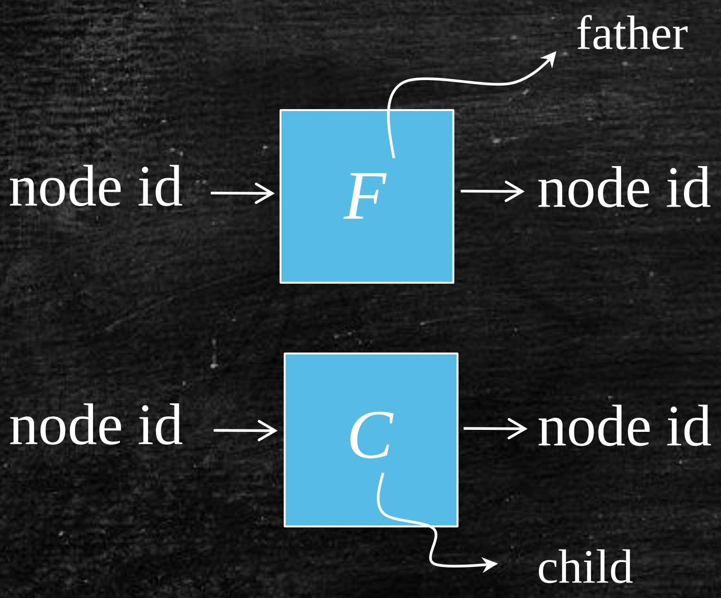
Parity Argument
Graph Representation

- Every vertex has in and out degree at most 1
- Each vertex with degree 1 is an acceptable solution (except the artificial one)

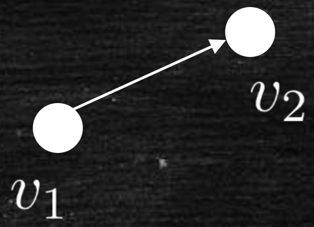


- By the parity argument there is always an even number of solutions
- Notice that if we insist in finding the pair of our green node the problem is beyond FNP!

The PPAD Class [Papadimitriou '94]



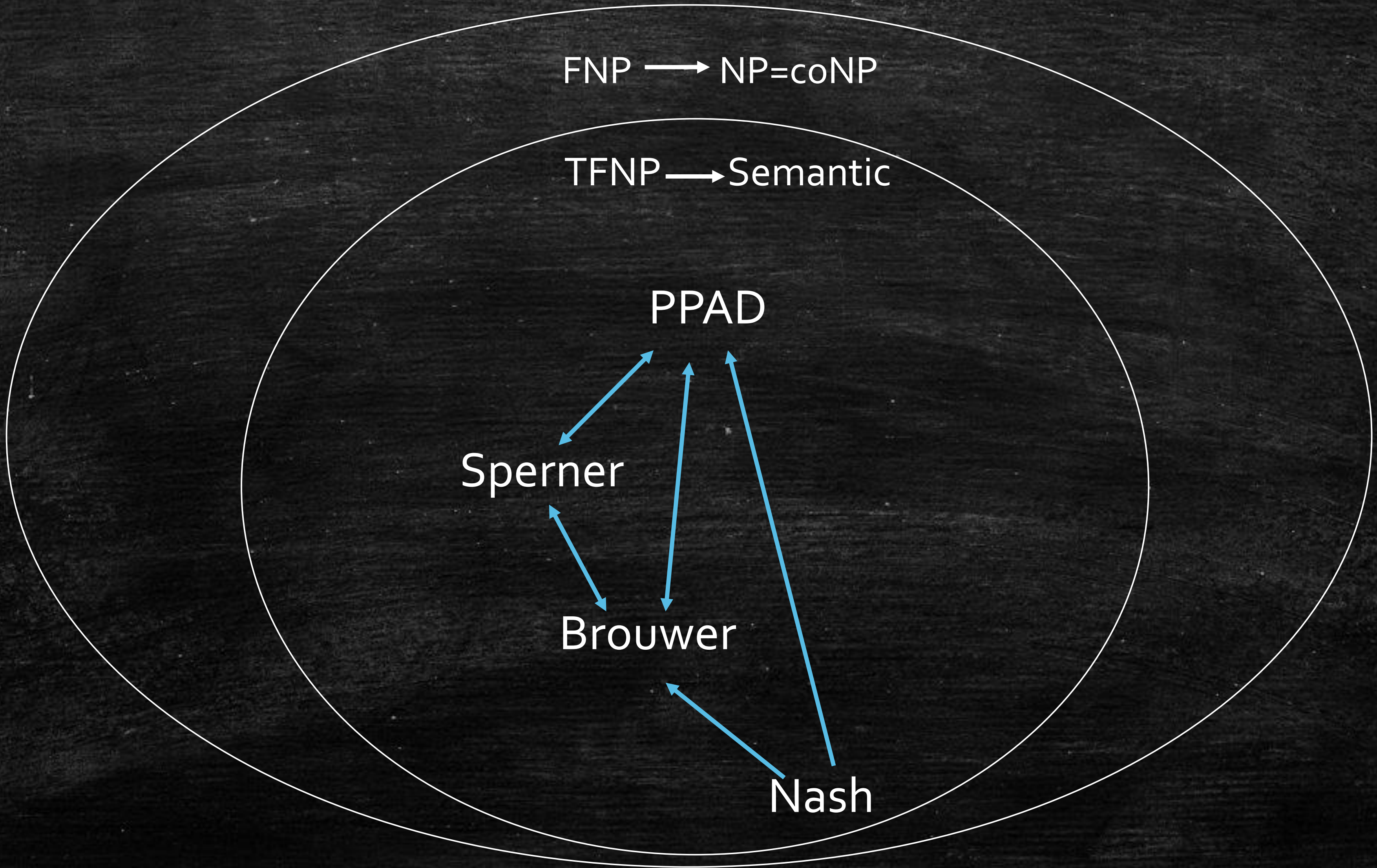
$$F(v_2) = v_1 \wedge C(v_1) = v_2$$



END OF THE LINE → Given F and C: If 0^n is an unbalanced node, find another unbalanced node. Otherwise say "yes".

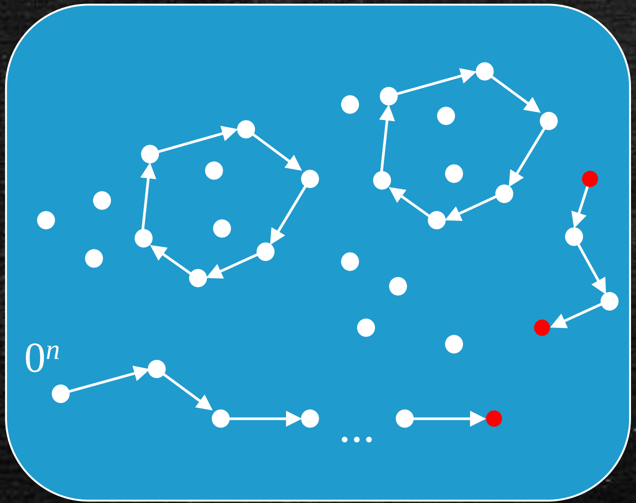
PPAD = { *Search problems in FNP reducible to END OF THE LINE* }

What we know



2-Nash PPAD-complete

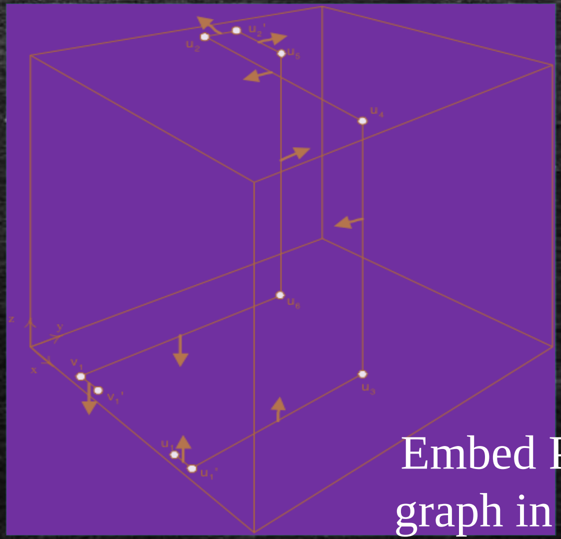
[Daskalakis,Goldberg,Papadimitriou 2006]



Generic PPAD

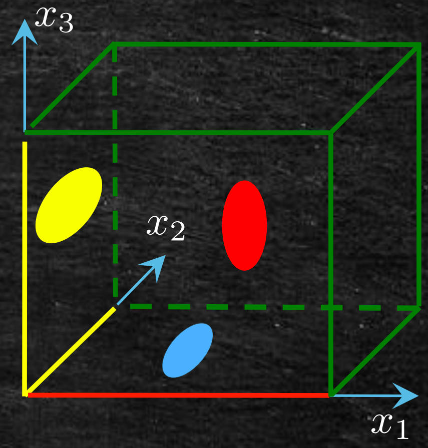
[Pap '94]

[DGP '05]



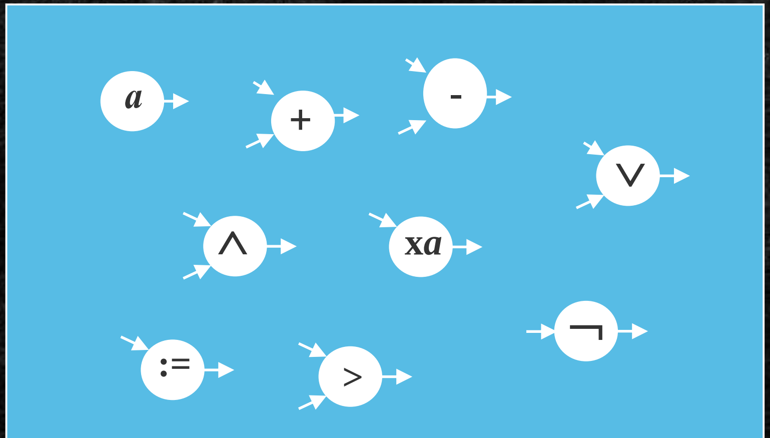
Embed PPAD graph in $[0,1]^3$

[DGP '05]



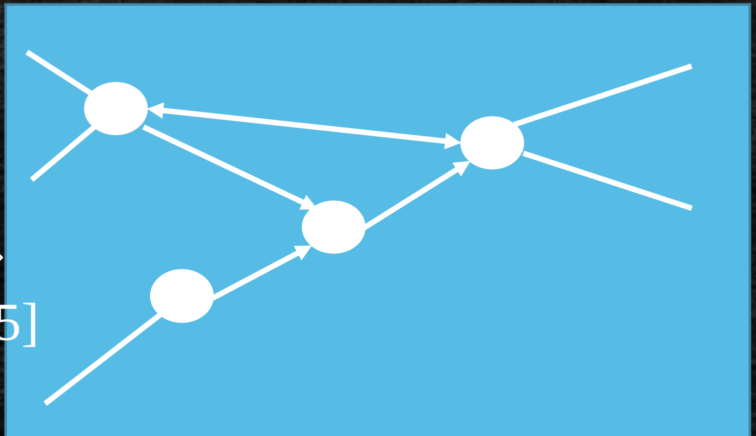
3D-SPERNER

[DGP '05]



Arithmetic Circuit Sat

[DGP '05]

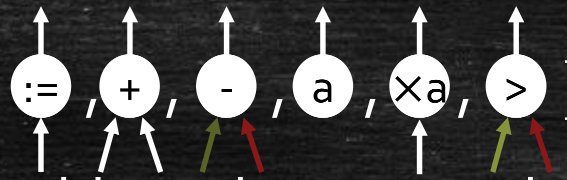


Polymatrix game

[DGP '05]

Arithmetic Circuit Sat

- Input A circuit with :

  - Variable nodes $x_1, x_2, \dots, x_n$
  - Gate nodes $g_1, g_2, \dots, g_m \in$ { 

  }
  - Directed edges connecting variable with gates and vice versa (loops are allowed)

- Output An assignment of values $x_1, x_2, \dots, x_n \in [0,1]$ 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 to const : $y == \max\{0, \min\{a, 1\}\}$

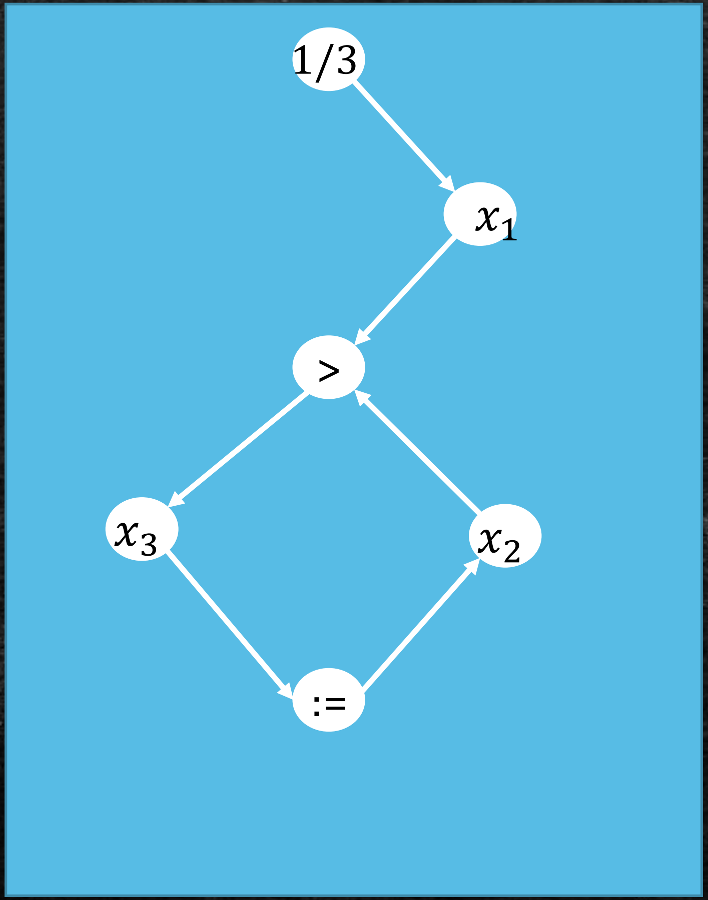
Multiply const: $y == \max\{0, \min\{ax, 1\}\}$

Arithmetic Circuit Sat

Comparison gate:

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

- Example :



Unique solution:

$$x_1 = x_2 = x_3 = \frac{1}{3}$$

Arithmetic Circuit Sat

- We can get an approximate version of Arithmetic Circuit Sat, by relaxing the gate constraints by $\epsilon \geq 0$:

Assignment : $y == x_1 \pm \epsilon$

Addition : $y == \min\{1, x_1 + x_2\} \pm \epsilon$

Subtraction : $y == \max\{0, x_1 - x_2\} \pm \epsilon$

Set to const : $y == \max\{0, \min\{a, 1\}\} \pm \epsilon$

Multiply const: $y == \max\{0, \min\{ax, 1\}\} \pm \epsilon$

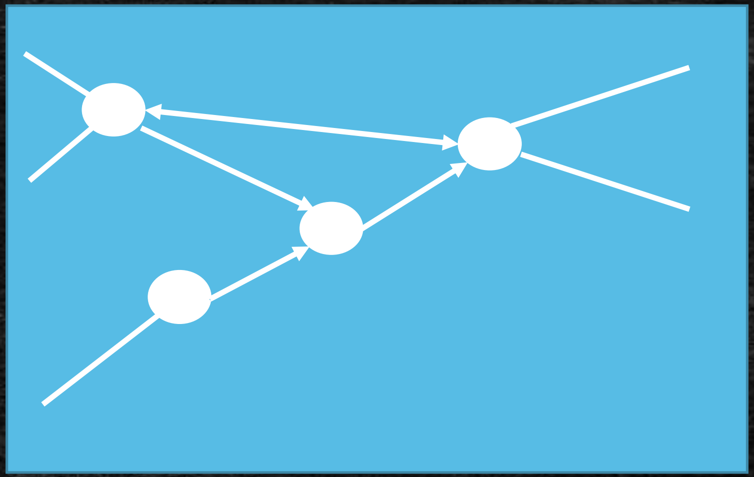
Comparison gate:

$$y == \begin{cases} 1, & if \;\; x_1 > x_2 - \epsilon \\ 0, & if \; x_1 < x_2 + \epsilon \\ *, & if \; x_1 = x_2 \pm \epsilon \end{cases}$$

Both versions of the problem are PPAD-complete!

Graphical Games

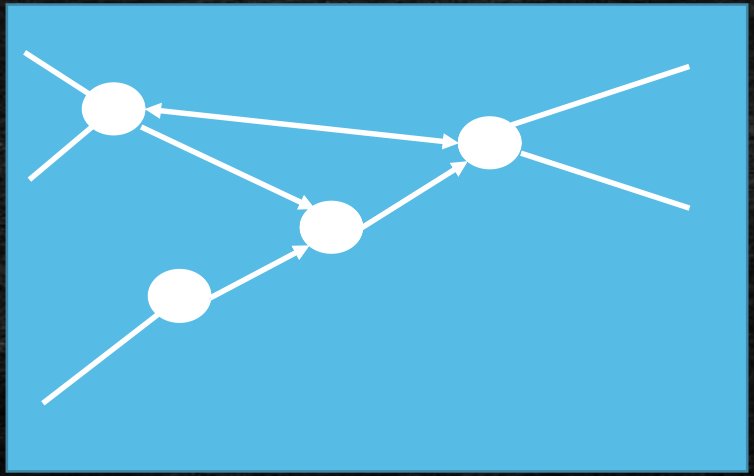
- Players are nodes in a directed graph.
- The player's payoff u_i depends on her strategy as well as the strategies of the players pointing to her.



Polymatrix Games

- Special case of Graphical Games.
- Payoff is edge-wise separable:

$$u_v(x_1, x_2, \dots, x_n) = \sum_{(w,v)\in E} u_{w,v}(x_w, x_v)$$



Arithmetic Circuit Sat → Polymatrix Game

- In order to reduce Arithmetic Circuit Sat to Polymatrix Games, we will present polymatrix gadgets which simulate the arithmetic functions of the circuit.

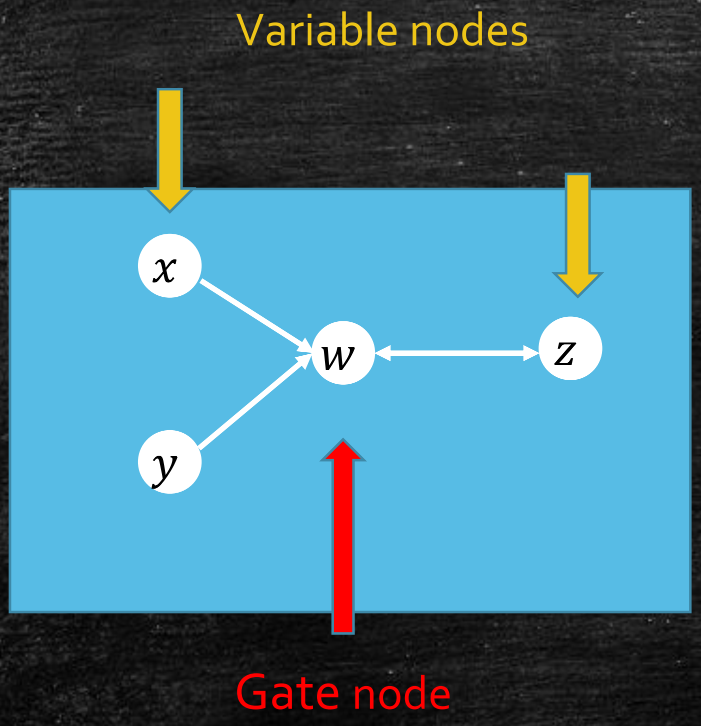
- Every player chooses her strategy from $\{0,1\}$.

- For every player p, representing a variable node x_p, $\Pr[p:1]$ represents the value of x_p.

- Finally, every Nash Equilibrium can be translated to a feasible solution of Arithmetic Circuit Sat.

Arithmetic Circuit Sat → Polymatrix Game

Addition Gadget



$$u(w\colon 0) = \Pr[x\colon 1] + \Pr[y\colon 1]$$

$$u(w\colon 1) = \Pr[z\colon 1]$$

$$u(z\colon 0) = 0.5$$

$$u(z\colon 1) = 1 - \Pr[w\colon 1]$$

In any Nash equilibrium of a game containing this Gadget $\quad \Pr[z\colon 1] = \min\{1, \Pr[x\colon 1] + \Pr[y\colon 1]\}$

Arithmetic Circuit Sat → Polymatrix Game
Addition Gadget

$u(w\colon 0) = \Pr[x\colon 1] + \Pr[y\colon 1]$

$u(w\colon 1) = \Pr[z\colon 1]$

$u(z\colon 0) = 0.5$

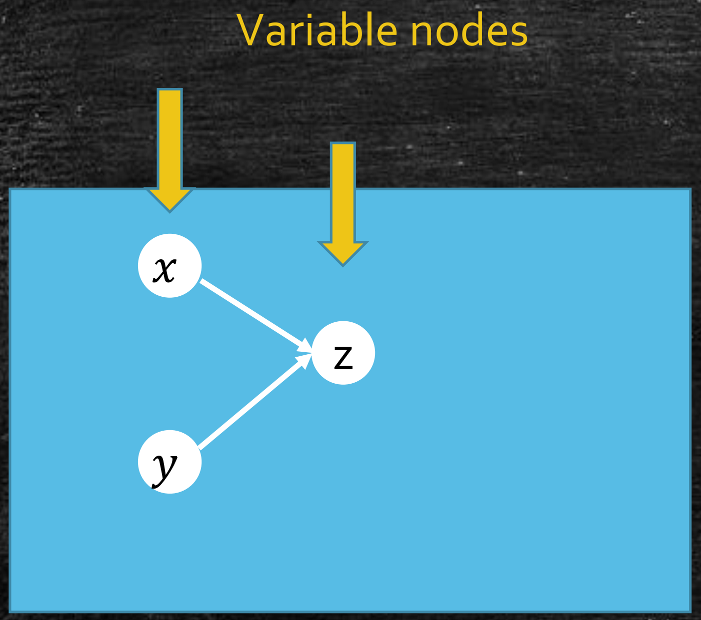
$u(z\colon 1) = 1 - \Pr[w\colon 1]$

- $\Pr[z\colon 1] < \min\{1, \Pr[x\colon 1] + \Pr[y\colon 1]\} \Rightarrow \Pr[w\colon 0] = 1 \Rightarrow \Pr[z\colon 1] = 1$
- $\Pr[z\colon 1] > \Pr[x\colon 1] + \Pr[y\colon 1] \Rightarrow \Pr[w\colon 1] = 1 \Rightarrow \Pr[z\colon 0] = 1$

⇓

$$\Pr[z\colon 1] = \min\{1, \Pr[x\colon 1] + \Pr[y\colon 1]\}$$

Arithmetic Circuit Sat → Polymatrix Game
Comparison Gadget



$u(z{:}\,0) = \Pr[y{:}\,1]$

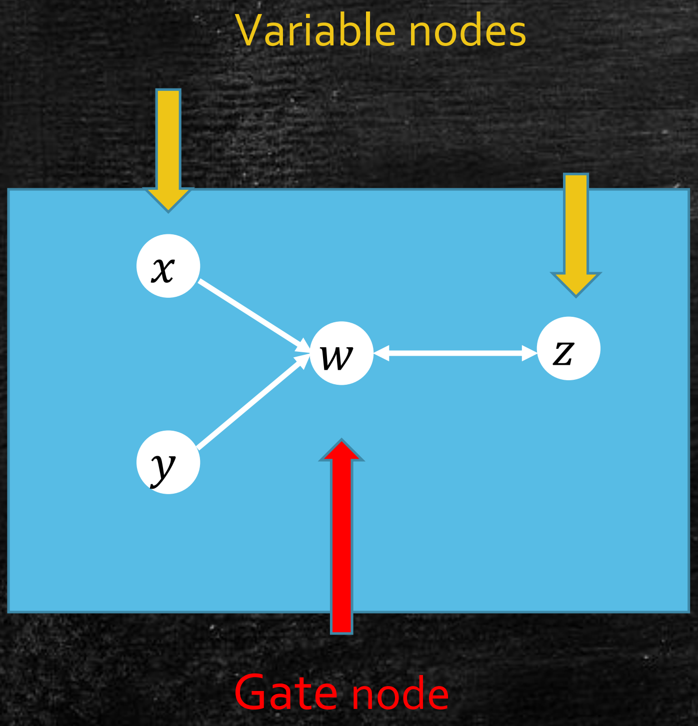
$u(z{:}\,1) = \Pr[x{:}\,1]$

$\Pr[x{:}\,1] > \Pr[y{:}\,1] \Rightarrow \Pr[z{:}\,1] = 1$

$\Pr[x{:}\,1] < \Pr[y{:}\,1] \Rightarrow \Pr[z{:}\,1] = 0$

$\Pr[x{:}1] = \Pr[y{:}\,1]$ anything is possible

From Polymatrix Game to 2-player Game



Every gadget can be turned into a bipartite graph with variable node-players sharing the same side and gate node-players on the other.

We define a 2-player game where the yellow lawyer represents all the yellow players and similarly the red lawyer represents all the red players.

The Lawyer Game

Our goal :

If (x, y) is a Nash Equilibrium for the Lawyer Game, then the marginal distributions that x assigns to the strategies of the yellow nodes and the marginal distributions that y assigns to the red nodes comprise a Nash Equilibrium in the Polymatrix Game.

In order to analyze the Lawyer Game we will first define and analyze two games, that combined will give us the appropriate game.

Breaking down the Lawyer Game

- The Representation Game:

The set of strategies for the yellow lawyer is the union of the strategies of every yellow node. The same goes for the red lawyer.

The payoff for the lawyers is the payoff that their clients would had gotten had they played the same strategies themselves.

- The High Stakes Chase

The sets of strategies remain the same.

Image an arbitrary labelling $\{1,..,n\}$ for the yellow clients and a respective labelling $\{1,...,n\}$ for the red clients.

Whenever both lawyers get to pick the same label, the red lawyer pays M to the yellow.

Otherwise they both get 0.

The Lawyer Game

The High Stakes Chase

Strategies of red node i

Strategies of yellow node j

M,-M	0, 0	0, 0	0, 0
0, 0	M,-M	0, 0	0, 0
0, 0	0, 0	M,-M	0, 0
0, 0	0, 0	0, 0	M,-M

It is easy to see that the High Stake Chase is a zero-sum game where in every NE the lawyers play uniformly over their clients.

Given this observation we could claim

Proposition 1:
Taking M arbitrarily big would essentially lead the lawyers to play with probability (approximately) $1/n$ each of their clients in the Combined Game!

(We no longer have to worry about our marginal distributions being ill-defined)

The Lawyer Game

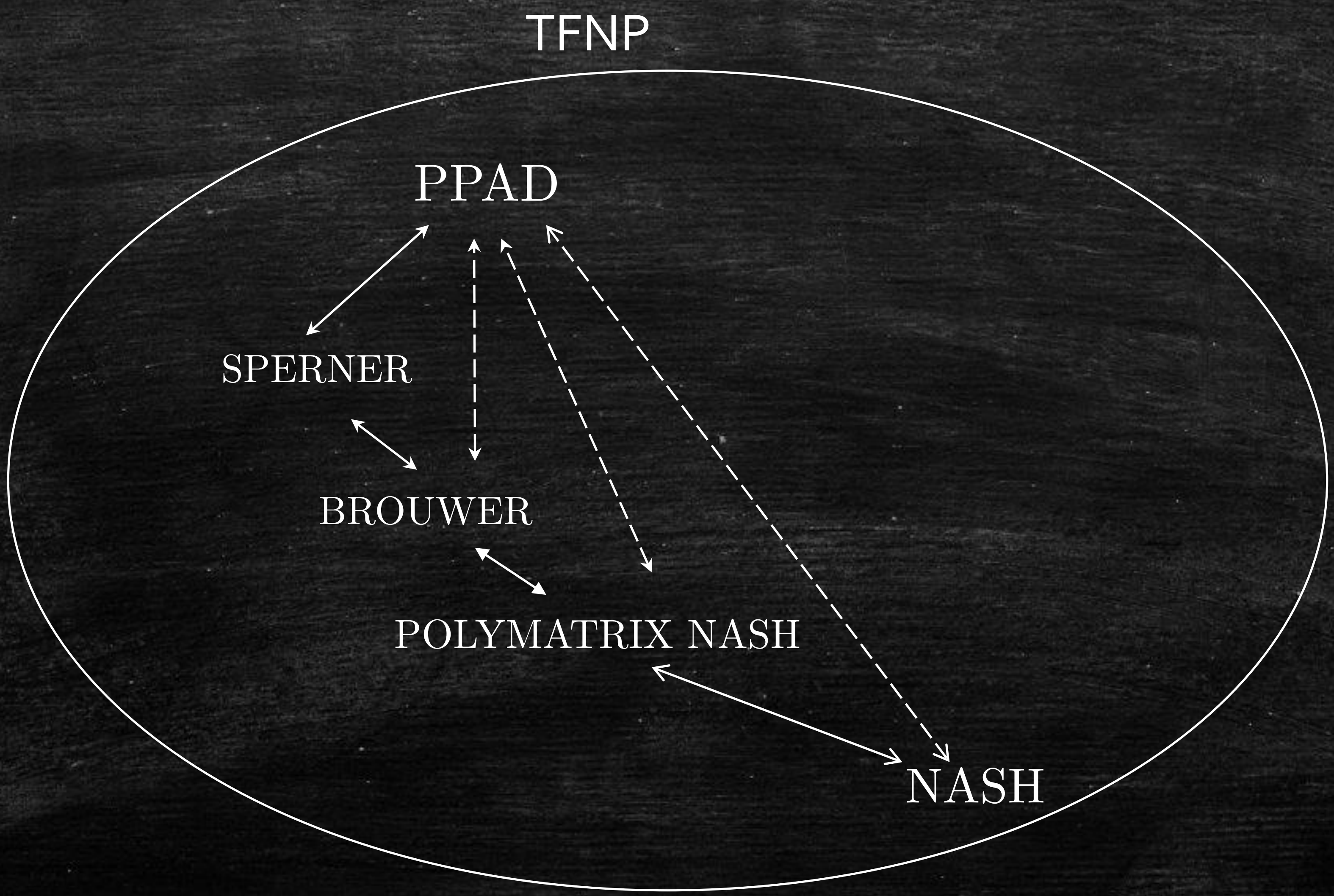
The Representation Game

On the other hand if both lawyers play uniformly over their clients, the way that the probability is split among each client's strategies will not affect the High Stakes Game.

The split will be solely determined by the Representation Game and this directly implies that our marginal distributions are indeed a NE for the Polymatrix Game.

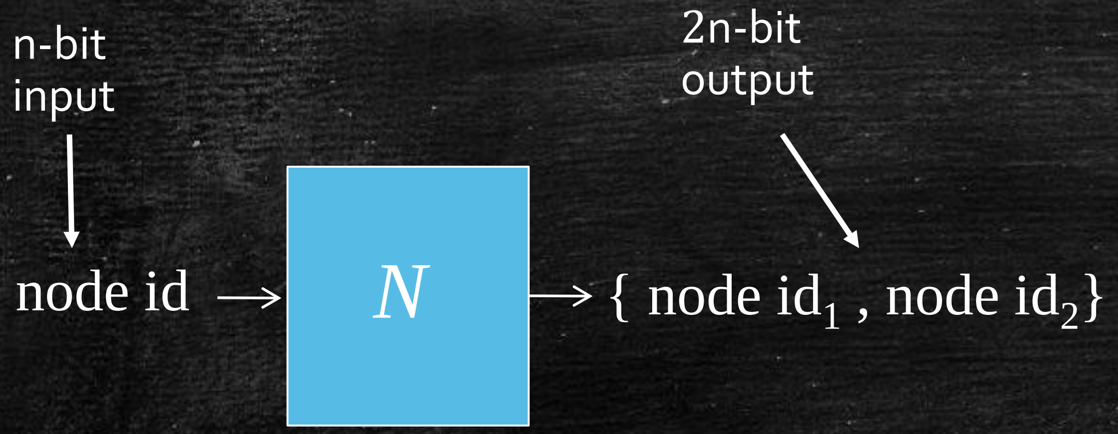
Notice that we are being a little bit inaccurate as Proposition 1 holds up to an error ϵ, but the sketch remains the same and the error can be accommodated by the Approximate Arithmetic Circuit Sat!

PPAD completeness of 2-Nash

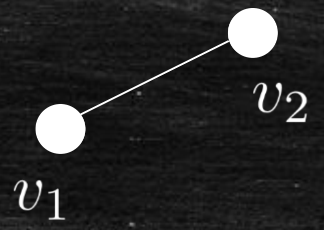


Arguments of existence and respective complexity classes
PPA[Papadimitriou '94]

'If a node has odd degree then there must be an other.'



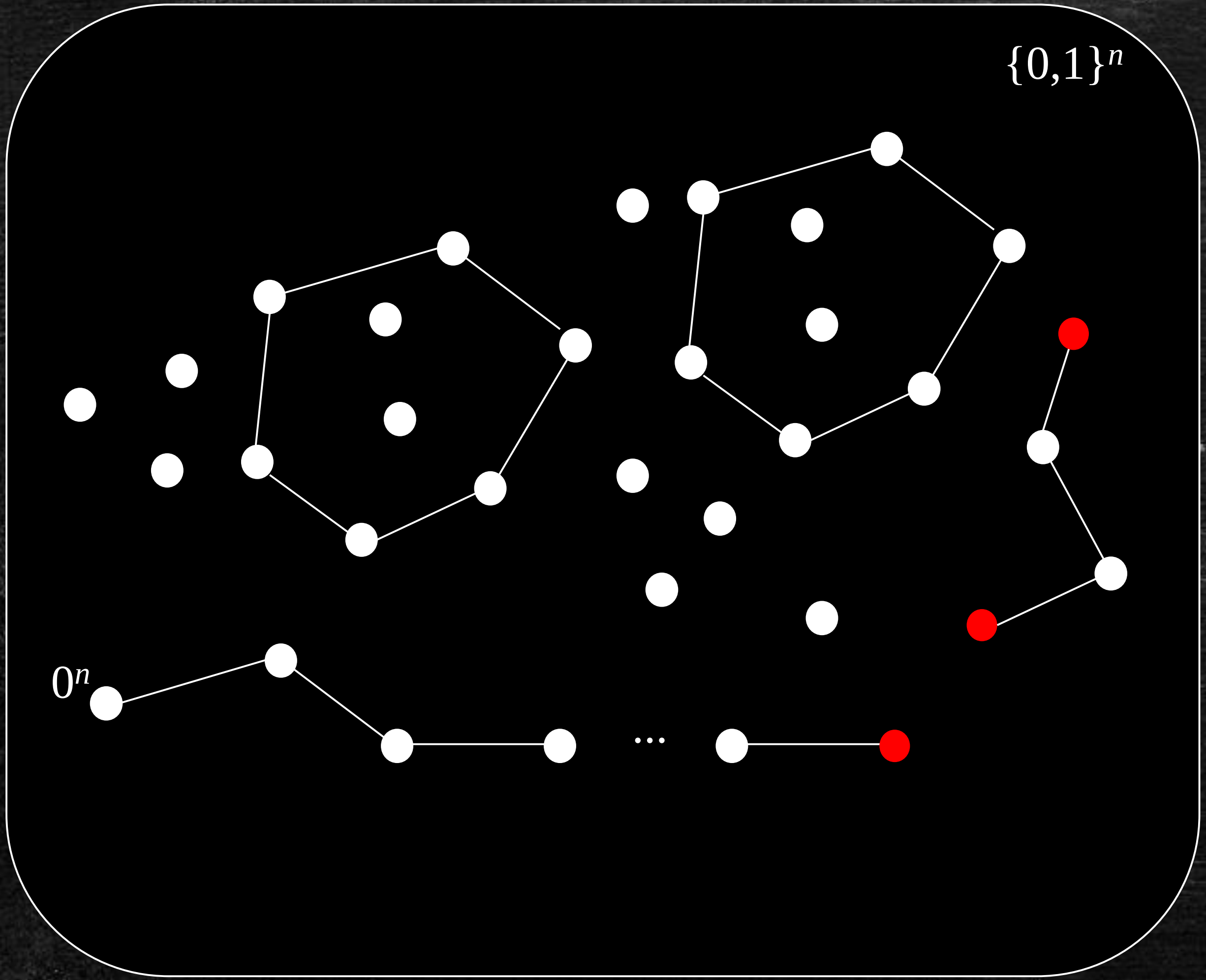
$v_1 \in N(u_2)$ & $u_2 \in N(u_1)$



ODD DEGREE NODE ⟶ Given N: If 0^n has odd degree, find another node with odd degree. Otherwise say "yes".

PPA = { *Search problems in FNP reducible to* ODD DEGREE NODE}

PPA
Graph Representation



Exponentially large graph

Every node has degree at most 2

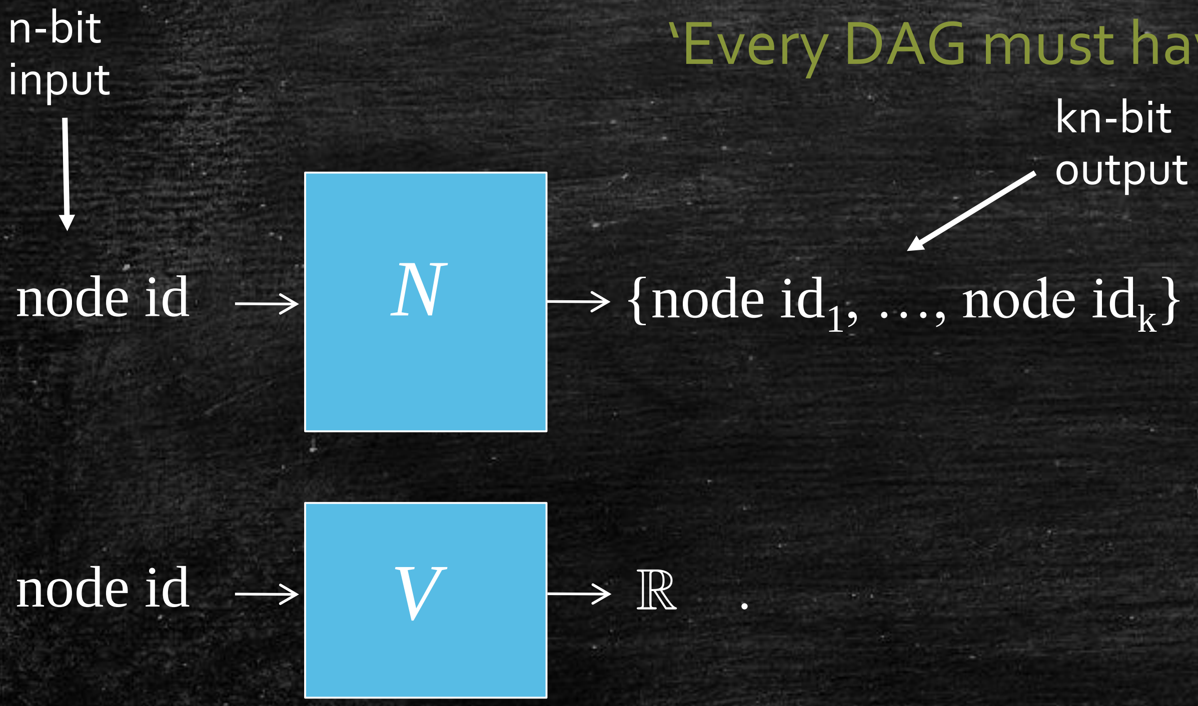
PLS [JPY '89]

'Every DAG must have a sink.'

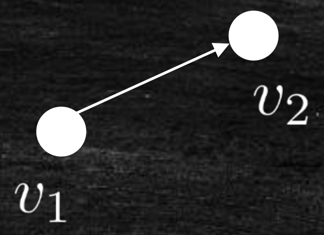
- **Local Max Cut** is a well known PLS-complete problem.
- **Spoiler! PNE** in **Congestion Games** is also PLS-complete.

PLS [JPY ‘89]

‘Every DAG must have a sink.’



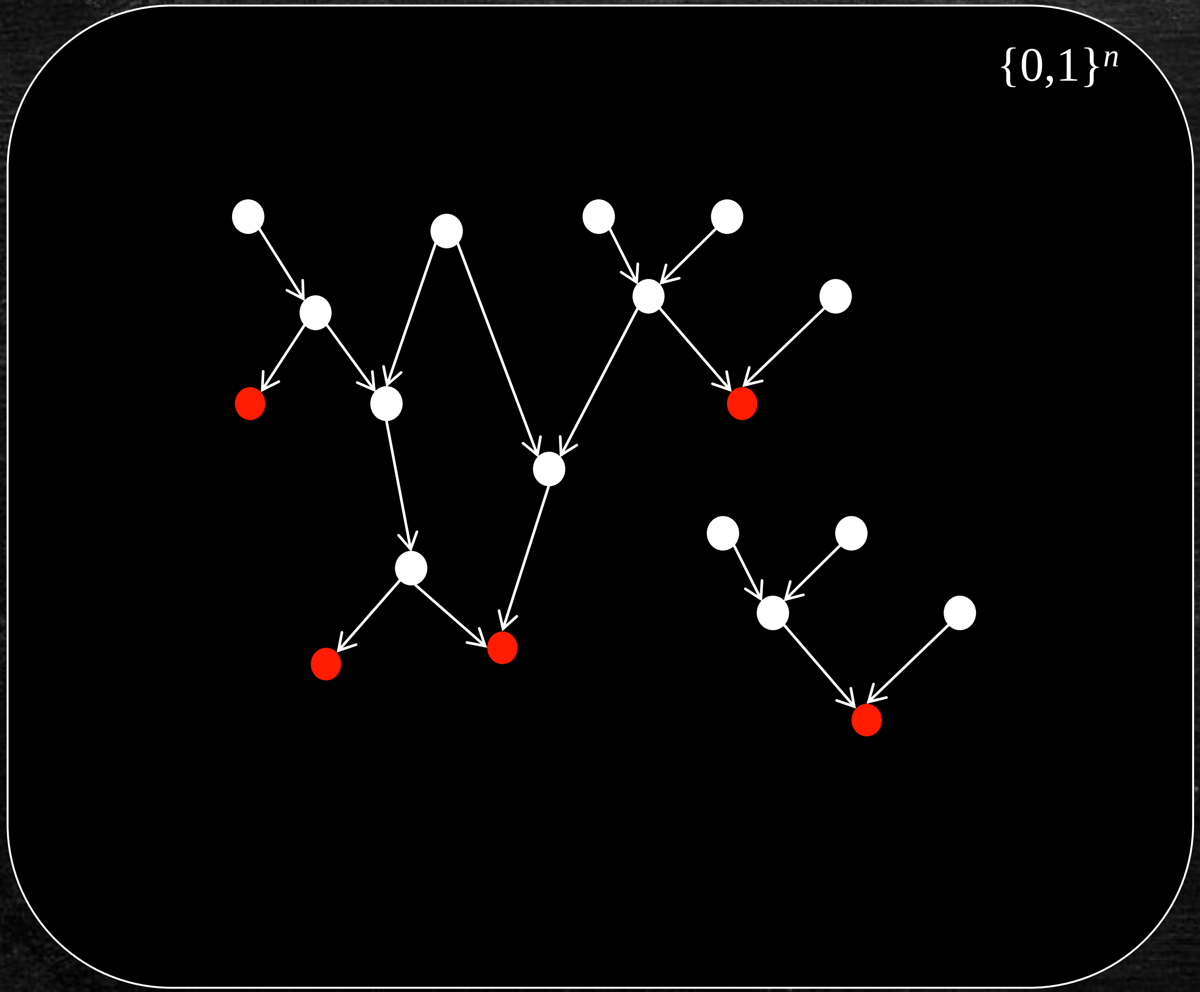
$$v_2 = N(v_1) \;\&\; V(v_2) > V(v_1)$$



FIND SINK ⟶ Given N, V: Find x s.t. $V(x) \geq V(y)$, for all $y \in N(x)$.

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

PLS
Graph Representation



Exponentially large directed acyclic graph

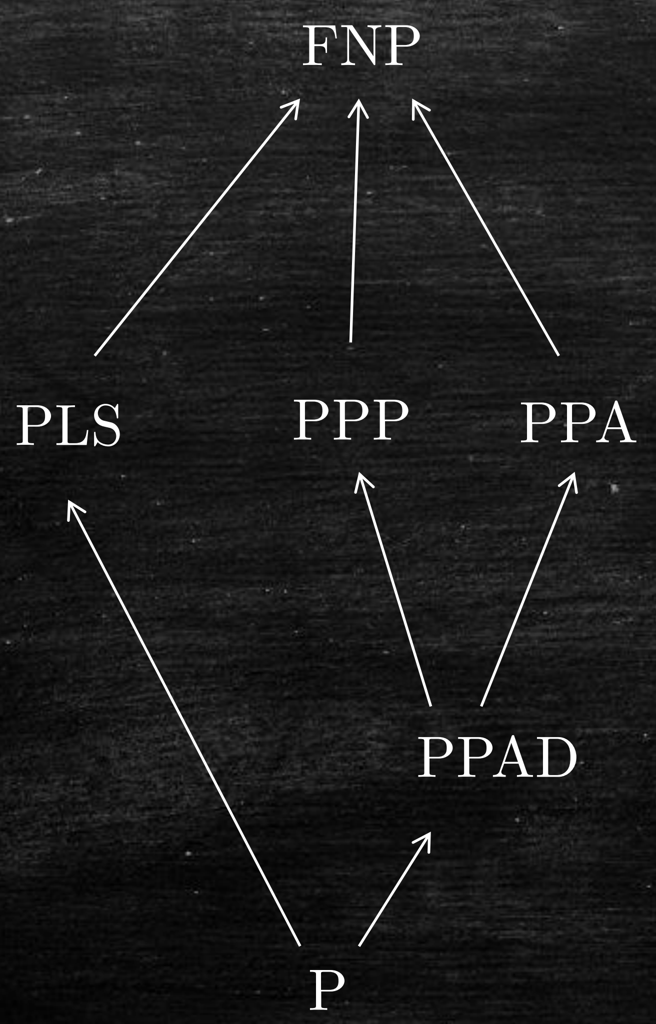
PPP

"If a function maps n elements to n-1 elements, then there is a collision."

COLLISION ⟶ Given F: Find x s.t. $F(x) = 0^n$; or find $x \neq y$ s.t. $F(x)=F(y)$.

PPP = { *Search problems in FNP reducible to* COLLISION }

Inclusions



$2-$player Symmetric Games

A bimatix game represented by two matrices (A, B) is called **Symmetric** if $B = A^T$ (i.e., the two players have the same set of strategies and their utilities remain the same if their roles are reversed).

A strategy profile x is a **Symmetric Nash Equilibrium** if both players playing x results in a Nash Equilibrium.

Looking at Symmetric Games is no loss of generality!

Reduction from Nash to Symmetric Nash

Fix any bimatrix game represented by the matrices A, B (w.l.o.g. with positive entries).
Now consider the Symmetric Game defined by the matrices below:

$$C_1 = \begin{array}{|c|c|} \hline \mathbf{0} & \boldsymbol{A} \\ \hline \boldsymbol{B}^{T} & \mathbf{0} \\ \hline \end{array} \begin{array}{l} \} \, x \\ \} \, y \end{array}$$

$$C_2 = \begin{array}{cc} x & y \\ \hline \multicolumn{1}{|c|}{\mathbf{0}} & \multicolumn{1}{c|}{\boldsymbol{B}} \\ \hline \multicolumn{1}{|c|}{\boldsymbol{A}^{\mathrm{T}}} & \multicolumn{1}{c|}{\mathbf{0}} \\ \hline \end{array}$$

Let (x, y) be a Symmetric NE.
In order (x, y) to be a best response to itself, x must be a best response to y and vice versa.

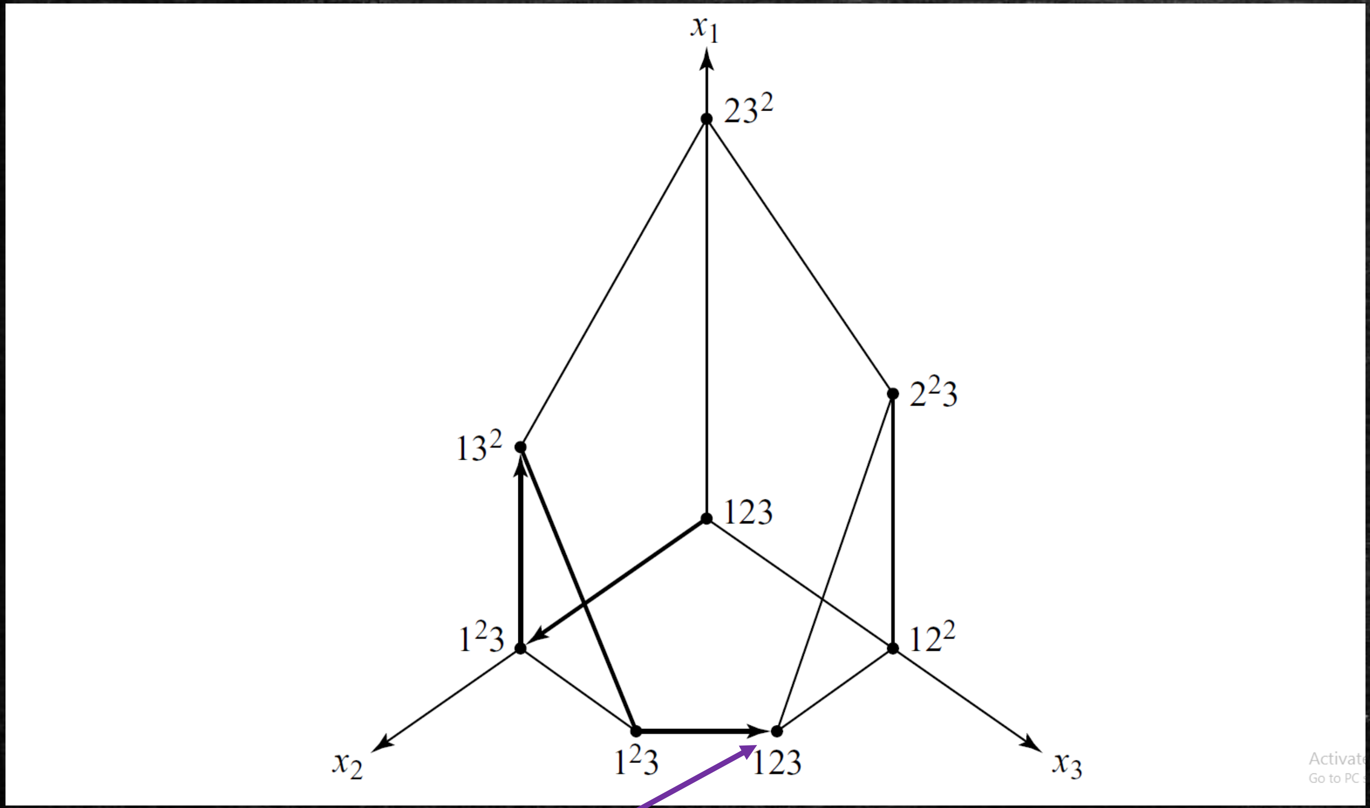
Lemke-Howson

- Fix any Symmetric Game with an, $n \times n$, utility matrix A.
- W.l.o.g. assume non negative entries and no zero column or raw.
- Consider the (non degenerate) polytope P defined by the following inequalities:

$$z \geq \mathbf{0}$$

$$Az \leq \mathbf{1} \text{ (!)}$$

Lemke-Howson



This vertex represents every strategy
It follows that here we get a SNE

1. A strategy i is represented at vertex z if $z_i = 0$ or $A_i z = 1$ or both.

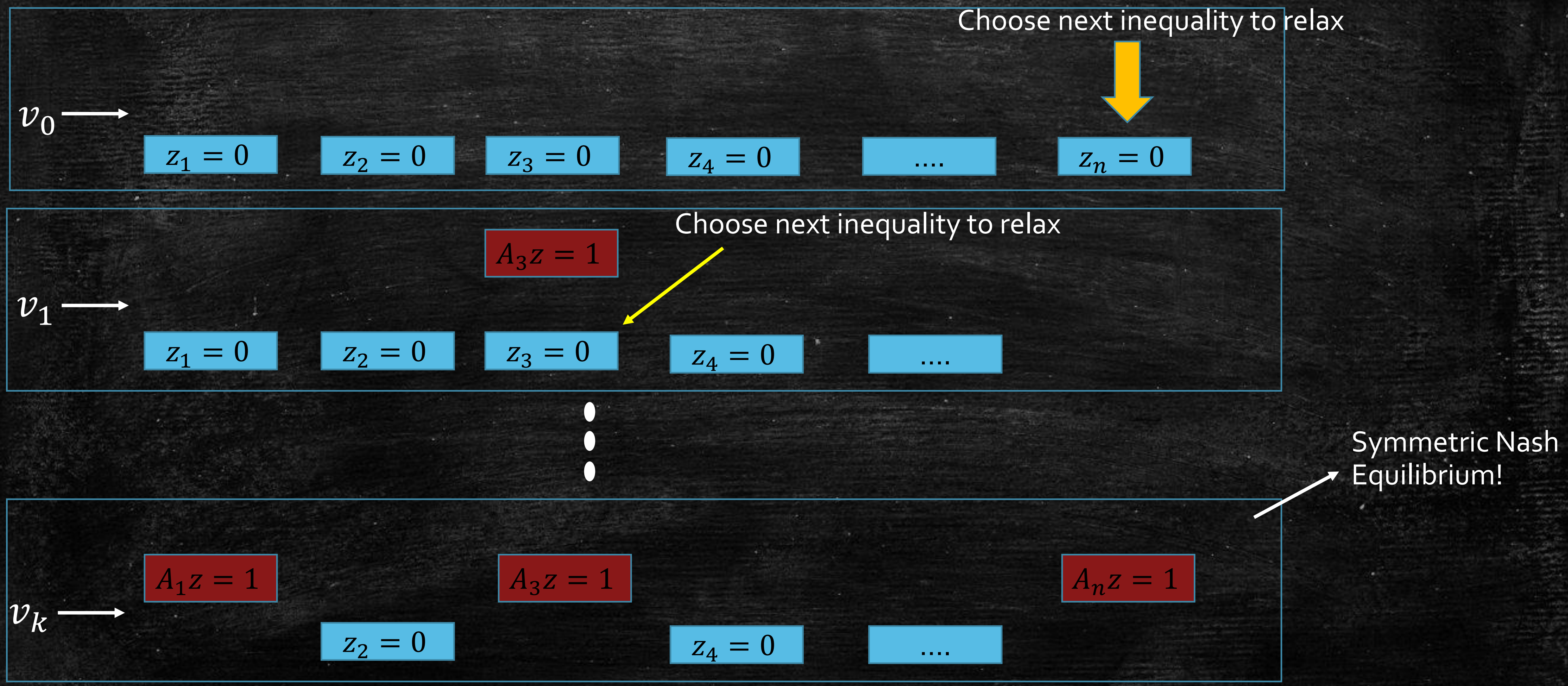
2. Define set V with all the vertices of P that represent every strategy except possibly strategy n

3. Any vertex (other than $\mathbf{0}$) at which all strategies are represented is a NE.

4. In order to find such a vertex we shall develop a (simplex-like) pivoting method beginning at vertex $\mathbf{0}$ and ending at a SNE.

$z_i = 0$ $A_i z = 1$

Lemke-Howson



Lemke-Howson

- Claim the walk can not loop neither can it reach the $\mathbf{0}$ vertex.(!)

- There are exponentially many but finite vertices in P.

It follows that the algorithm halts returning a SNE.

Final remark: Although it may seem like there are no direction in the edges we define, in fact this algorithm relates to PPAD.

Recap

GOOD NEWS

- NE in 2-player zero sum ↔ LP Duality

BAD NEWS

- NE in general 2-player games PPAD complete
  (Lemke-Howson exponential running time algorithm)

SOLUTION

In order to sidestep the probable intractability of NE we are going to relax our equilibrium concept!

Approximate Nash Equilibrium

For any $\varepsilon > 0$ a pair of mixed strategies x, y is called an ε -Nash equilibrium if:

i. For every mixed strategy x' of the row player, $(x', Ay) \leq (x, Ay) + \varepsilon$

ii. For every mixed strategy y' of the column player, $(x\ , By') \leq (x, By) + \varepsilon$

Lipton Markakis Mehta '03

Main result

(Assuming all entries of A, B between 0,1)

For any NE x^*, y^* and for any $\varepsilon > 0$, there exists, for every $k \geq {}^{12 \ln n}/_{\varepsilon^2}$ a pair of k –uniform strategies $x', y',$ such that:

1. x', y' is an ε-NE
2. $|(x', Ay') - (x^*, Ay^*)| < \varepsilon$ (row player gets almost the same payoff as in the NE)
3. $|(x', By') - (x^*, By^*)| < \varepsilon$ (column player gets almost the same payoff as in the NE)

Lipton Markakis Mehta '03
Proof Sketch via Probabilistic Method

- Given $x^*, y^*, \varepsilon > 0$ fix $k \geq {}^{12 \ln n}/_{\varepsilon^2}$

- Form multiset X sampling k times independently, from the pure strategies of the row player according to the distribution x^*.
  Respectively, form Y from the pure strategies of the column player.

- Let x' be the k −uniform strategy related with multiset X and y' the k −uniform strategy related with multiset Y.

Lipton Markakis Mehta '03

Proof Sketch via Probabilistic Method

- Finally consider the following events:

- $\varphi_1 = \{|(x', Ay') - (x^*, Ay^*)| < \frac{\varepsilon}{2}\}$
- $\varphi_2 = \{|(x', By') - (x^*, By^*)| < \frac{\varepsilon}{2}\}$
- $\pi_{1,i} = \{(x^i, Ay') - (x', Ay') < \varepsilon\}$ $(i = 1,2 \ldots, n)$
- $\pi_{2,j} = \{(x', By^j) - (x', By') < \varepsilon\}$ $(j = 1,2 \ldots, n)$

$$GOOD = \varphi_1 \cap \varphi_2 \cap_{i=1}^{n} \pi_{1,i} \cap_{j=1}^{n} \pi_{2,j}$$

Goal: $\Pr[GOOD^c] < 1$

Lipton Markakis Mehta '03

Proof Sketch via Probabilistic Method

In order to bound the probability of φ_1^c we define the following :

$$\varphi_{1a} = \{|(x', Ay^*) - (x^*, Ay^*)|\} < \frac{\varepsilon}{4}$$
$$\varphi_{1b} = \{|(x', Ay') - (x', Ay^*)|\} < \frac{\varepsilon}{4}$$

The expression (x', Ay^*) is a sum of k independent random variables each of expected value (x^*, Ay^*). Each such random variable takes value between 0 and 1. As a result we can apply Chernoff bounds:

$$\Pr[\varphi_{1a}^c] \leq 2e^{-\frac{k\varepsilon^2}{8}}$$

and similarly

$$\Pr[\varphi_{1b}^c] \leq 2e^{-\frac{k\varepsilon^2}{8}}$$

$$\varphi_{1a} \cap \varphi_{1b} \subseteq \varphi_1 \quad \Rightarrow \quad \Pr[\varphi_1^c] \leq \Pr[\varphi_{1a}^c \cup \varphi_{1b}^c] \leq 4e^{-\frac{k\varepsilon^2}{8}}$$

Lipton Markakis Mehta '03
Proof Sketch via Probabilistic Method

Using the same toolbox we get the following bounds:

$$\Pr[\varphi_1^c] \leq 4e^{-\frac{k\varepsilon^2}{8}}$$

$$\Pr[\varphi_2^c] \leq 4e^{-\frac{k\varepsilon^2}{8}}$$

$$\Pr\left[\pi_{1,i}^c\right] \leq 4e^{-\frac{k\varepsilon^2}{8}} + 2e^{-\frac{k\varepsilon^2}{2}}$$

$$\Pr\left[\pi_{2,j}^c\right] \leq 4e^{-\frac{k\varepsilon^2}{8}} + 2e^{-\frac{k\varepsilon^2}{2}}$$

$$\Pr[GOOD^c] \leq \Pr[\varphi_1^c] + \Pr[\varphi_2^c] + \sum_{i=1}^{n} \Pr\left[\pi_{1,i}^c\right] + \sum_{j=1}^{n} \Pr\left[\pi_{1,j}^c\right]$$

$$\leq 8e^{-\frac{k\varepsilon^2}{8}} + 2n\left(e^{-\frac{k\varepsilon^2}{2}} + 4e^{-\frac{k\varepsilon^2}{8}}\right) < 1$$

Lipton Markakis Mehta ’03
Subexponential running time & m −player games

The main result implies the existence of subexponential algorithm ($n^{O(logn)}$) for computing all k −uniform ε − equilibria for any 2 −player game(!)

The main result can accommodate m −player games although the dependence of k to m is polynomial.

Barman’s sparsification technique ‘14

Applying the approximate version of Caratheodory’s theorem Barman improved the previous results proving the following statement:

(Assuming all entries of A, B between $0,1$)

In any bimatrix game with $n \times n$ matrices A, B, if the number of non-zero entries in any column of $A + B$ is at most s then an ε −NE can be computed in time

$$n^{O\left(logs/\varepsilon^2\right)}$$

Anonymous Games

In **Anonymous Games** a large population of players shares the same strategy set and, while players may have different payoff functions, the payoff of each depends on her own choice of strategy and the number of the other players playing each strategy (not the identity of these players).

Canonical example:
500 citizens have to decide either to go to the cinema or to the theatre and they only care about how crowded it will be.

Polynomial-Time Approximation Scheme (PTAS)

A PTAS is an algorithm which takes an instance of an optimization problem and a parameter $\varepsilon > 0$ and, in polynomial time, produces a solution that is within a factor $1 + \varepsilon$ of being optimal.

Notice that an algorithm running in time $O\left(n^{\varepsilon^{-1}}\right)$ or even $O(n^{\exp(\varepsilon^{-1})})$ counts as a PTAS.

PTAS for Anonymous Games
(Daskalakis, Papadimitriou ’14)

There is a PTAS for the mixed Nash equilibrium problem for normalized anonymous games with a constant number of strategies.
More precisely, there exists some function g such that, for all $\varepsilon > 0$, an ε-Nash equilibrium of a normalized anonymous game of m players and n strategies can be computed in time $m^{g(n,\varepsilon^{-1})}$.

Wrapping up

- Computing exact NE in $2-$player zero sum games belongs in P
- Computing exact NE in general $2-$player games is PPAD complete
- Computing $\varepsilon-$NE in general $2-$player games accepts subexponential time algorithms
- Computing $\varepsilon-$NE in Anonymous Games accepts PTAS algorithms


Thank You!