(guest lecture by Costis Daskalakis)

## Recall:

- existence theorems: Nash, Brouwer, Sperner
- → total NP search problems
- Parity Arguments in Directed graphs (PAD)
- PPAD = class of Problems reducible to:
- End of the Line
  - circuits P & N: maps on $n$-bit node ids
  - edge $(v, w) \iff N(v) = w \wedge P(w) = v$
  - if given vertex is degree 1, find another
- Arithmetic Circuit SAT
  - circuit with cycles, no inputs, arbitrary fanout
  - total & PPAD-complete

## Graphical games: [Kearns, Littman, Singh 2001]

- motivation: geographic/otherwise limited interaction
- players = nodes in a graph
- payoff depends only on your own & neighbors' strats.
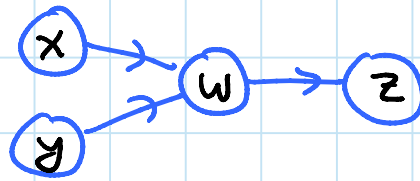
## Polymatrix games: [Janovskaya 1968]

- graphical games with edge-wise separable utility functions:

$$u_{w,v}(x_v, x_w) = \mathop{\mathbb{E}}_{\substack{s_u \sim x_u \\ s_w \sim x_w}} u_{w,v}(s_v, s_w) = \sum_{s_u, s_w} u_{w,v}(s_v, s_w) \cdot x_v(s_v) \cdot x_w(s_w)$$

mixed strategies

<u>PPAD-completeness of Nash:</u>   [Daskalakis, Goldberg,
- reduction from Arithmetic Circuit SAT   Papadimitriou
- each player's strategy $\in \{0, 1\}$                          2006]
$\Rightarrow$ mixed strategy $\in [0, 1]$
- addition gadget:

$x \rightarrow w \rightarrow z$
$y \rightarrow$

- w paid expected $\begin{cases} \$ \Pr\{x:1\} + \Pr\{y:1\} & \text{if plays } 0 \\ \$ \Pr\{z:1\} & \text{if plays } 1 \end{cases}$

via matrix

| w:0 | y:0 | y:1 |
|-----|-----|-----|
| x:0 | 0 | 1 |
| x:1 | 1 | 2 |

| w:1 | z:0 | z:1 |
|-----|-----|-----|
|  | 0 | 1 |

- z is paid to play opposite of w:
$$u(z:0) = 0.5 \qquad u(z:1) = 1 - \Pr\{w:1\}$$
$\Rightarrow$ in any Nash equilibrium:
$$\Pr[z:1] = \min\{\Pr[x:1] + \Pr[y:1], 1\}$$
$- \Pr[z:1] < \min\{\Pr[x:1] + \Pr[y:1], 1\}$
$\Rightarrow \Pr[w:0] = 1$
$\Rightarrow \Pr[z:1] = 1$  ✖  contradiction
$- \Pr[z:1] > \Pr[x:1] + \Pr[y:1]$
$\Rightarrow \Pr[w:1] = 0$
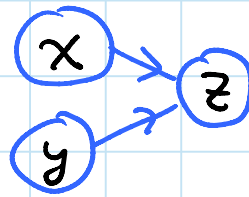$\Rightarrow \Pr[z:1] = 0$ ✖ contradiction
- subtraction gadget: just change:

|  | y:0 | y:1 |
|-----|-----|-----|
| x:0 | 0 | -1 |
| y:1 | 1 | 0 |

— comparison gadget:



|      | y:0 | y:1 |
|------|-----|-----|
| z:0  | 0   | 1   |

|      | x:0 | x:1 |
|------|-----|-----|
| z:1  | 0   | 1   |

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

— etc. $\rightarrow$ polymatrix game

$\Rightarrow$ Arithmetic Circuit SAT is total (via Nash theorem) with rational, polynomial solution complexity solution

<u>Polymatrix game $\rightarrow$ 2-player game</u>: [Chen & Deng 2006]
  — make previous game bipartite    (almost in DGP 2006)
  — one player ("lawyer") per color class
    — strategies = <u>union</u> of vertices' strategies
      $\hookrightarrow$ pure $\Longleftarrow$
    — payoffs only along edges of directed graphs
  — <u>hope</u>: Nash equilibrium of lawyer game
      $\Rightarrow$ marginal distributions are Nash equilibrium
         of polymatrix game
  — but lawyers prefer to represent lucrative clients
  — <u>fix</u>: additional high-stakes game
    — if red lawyer plays strategy for red vertex i
       & blue lawyer plays strategy for blue vertex j
       then — if $i \neq j$ then both get \$0   $\rightarrow 2n \cdot U_{max}$
            — if $i = j$ then red gets $+\infty$ & blue gets $-\infty$
  $\Rightarrow$ lawyers uniformly choose which vertex to rep.

- game = sum of two games
- in any Nash equilibrium:
  - close to uniform on vertices: $\forall u, v$
  $$x_u = \frac{1}{n}\left(1 \pm \frac{2 u_{max} n^2}{\infty}\right), \text{ ditto for } y_v$$
  - within vertex, payoff difference for red lawyer between $u{:}i$ & $u{:}j$ is $\sum_v \sum_\ell \left(A^{(u,v)}_{i,\ell} - A^{(u,v)}_{j,\ell}\right)$ <span style="color:green">(no $\infty$)</span>

$\Rightarrow$ if $x_{u:i} > 0$ then $\forall j: \sum_u \sum_\ell \left(A^{(u,v)}_{i,\ell} - A^{(u,v)}_{j,\ell}\right) \geq 0$
  - define marginals $\hat{x}_u(i) = \frac{x_{u:i}}{x_u}$ & $\hat{y}_v(j) = \frac{y_{v:j}}{y_v}$
  - get approximate Nash equilibrium (would be exact if uniform on players)
  - set $\infty$ large
  - all these problems were hard to $\varepsilon$-approximate

## Easy 2-player Nash: payoff matrices R & C
  - zero sum $\Longleftrightarrow$ R+C = $\emptyset$
    - polynomial time via LP
  - rank r $\Longleftrightarrow$ R+C has rank r
  - rank 1 $\Rightarrow$ poly. time
  - rank 3 $\Rightarrow$ PPAD-complete    <span style="color:purple">[Mehta 2014]</span>

OPEN: $\varepsilon$-Nash: $\leq$additive $\varepsilon$ incentive for player to change
  - no FPTAS
  - quasiPTAS: $n^{O\left(\frac{\lg n}{\varepsilon^2}\right)}$ time

PPA: "if a graph has a node of odd degree
     then it must have another" (Handshaking Lemma)
— undirected version of PPAD
— Odd Degree Node:
    — one circuit $C: V \rightarrow$ sets of 2 vertices
    — edge $(u,v) \Leftrightarrow v \in C(u) \wedge u \in C(v)$
    $\Rightarrow$ max. degree 2
    — if given node has odd degree,
      find another one
    — PPA-complete
— crucial here & in PPAD that we ask for
  some other vertex. not other end of same
  path $\sim$ else not in FNP
— Smith: given Hamiltonian cycle in 3-regular graph
         find another one
      $\in$ PPA
    — OPEN : PPAD-complete ?

explicit
always exists [Smith]
[Thomason 1978]

PLS: "every directed acyclic graph has a sink"

[Johnson, Papadimitriou, Yannakakis 1989]

- Find Sink: — n-bit node ids.
  - circuit $C: V \to 2^V$ → sets of vertices
  - circuit $F: V \to \mathbb{R}$
  - edge $(u, v) \Leftrightarrow v \in C(u) \wedge F(v) > F(u) \Rightarrow DAG$
  - goal: find a sink ($x$ s.t. $F(x) > F(v) \; \forall v$)
- PLS = {FNP problems reducible to Find Sink}
- Local Max Cut:
  - given a weighted graph
  - find a partition $V = V_1 \cup V_2$ such that can't move any one vertex $V_1 \leftrightarrow V_2$ to increase size of cut $E \cap (V_1 \times V_2)$
  - PLS-complete [Schaffer & Yannakakis 1991]
  - easy for unit/small weights

PPP: "if a function maps n elements to n-1 slots, then there is a collision" (Pigeonhole Principle)
= {FNP problems reducible to Collision}
- Collision: given circuit $C: V \to V$, find
  ① $x$ s.t. $C(x) = 0^n$   or  ② find $x \neq y$ s.t. $C(x) = C(y)$

Relationships:

$$FP \underset{\subseteq}{\overset{\subseteq}{\phantom{x}}} \begin{matrix} PPAD \overset{\subseteq}{\underset{\subseteq}{\phantom{x}}} \begin{matrix} PPP \\ PPA \end{matrix} \overset{\subseteq}{\underset{\subseteq}{\phantom{x}}} FNP \\ PLS \subseteq \end{matrix}$$