Equilibrium Computation in Normal Form Games

Costis Daskalakis & Kevin Leyton-Brown

Part 1(a)

Overview

1. Plan of this Tutorial
2. Getting Our Bearings: A Quick Game Theory Refresher
3. Solution Concepts
4. Computational Formulations

Plan of this Tutorial

- This tutorial provides a broad introduction to the recent literature on the computation of equilibria of simultaneous-move games, weaving together both theoretical and applied viewpoints.
- It aims to explain recent results on:
  - the complexity of equilibrium computation;
  - representation and reasoning methods for compactly represented games.
- It also aims to be accessible to those having little experience with game theory.
- Our focus: the computational problem of identifying a Nash equilibrium in different game models.
- We will also more briefly consider ϵ-equilibria, correlated equilibria, pure-strategy Nash equilibria, and equilibria of two-player zero-sum games.

Part 1: Normal-Form Games (2:00 PM – 3:30 PM)

Part 1a: Game theory intro *(Kevin)*

- Game theory refresher; motivation
- Game theoretic solution concepts
- Fundamental computational results on solution concept computation

Part 1b: Complexity of equilibrium computation *(Costis)*

- Key result: the problem of computing a Nash equilibrium is PPAD-complete
- The complexity of approximately solving this problem

Part 2: Compact Game Representations (4:00 PM – 5:30 PM)

Part 2a: Introducing compact representations *(Costis)*

- Foundational theoretical results about the importance and challenges of compact representation
- Symmetric games
- Anonymous games

Part 2b: Richer compact representations *(Kevin)*

- Congestion games
- Graphical games
- Action-graph games

Overview

Normal-Form Games

Normal-form games model simultaneous, perfect-information interactions between a set of agents.

Definition (Normal-Form Game)

A finite, n-person game $\langle N, A, u\rangle$ is defined by:

- N: a finite set of n players, indexed by i;
- $A = \langle A_1, \ldots, A_n\rangle$: a tuple of action sets for each player i;
  - $a \in A$ is an action profile
- $u = \langle u_1, \ldots, u_n\rangle$: a utility function for each player, where $u_i : A \mapsto \mathbb{R}$.

In a sense, the normal form is the most fundamental representation in game theory, because all other representations of finite games (e.g., extensive form, Bayesian) can be encoded in it.

Example Games

	Heads	Tails
Heads	$1, -1$	$-1, 1$
Tails	$-1, 1$	$1, -1$

- Matching Pennies:
  - agents choose heads and tails;
  - one agent wants to match and one wants to mismatch.

Example Games

	Heads	Tails
Heads	$1, -1$	$-1, 1$
Tails	$-1, 1$	$1, -1$

	B	F
B	$2, 1$	$0, 0$
F	$0, 0$	$1, 2$

- Matching Pennies:
  - agents choose heads and tails;
  - one agent wants to match and one wants to mismatch.
- Battle of the Sexes:
  - husband likes ballet better than football
  - wife likes football better than ballet
  - both prefer to be together

Mixed Strategies

- In some games (e.g., matching pennies) any deterministic strategy can easily be exploited
- Idea: confuse the opponent by playing randomly
- Define a strategy s_i for agent i as any probability distribution over the actions A_i.
    - pure strategy: only one action is played with positive probability
    - mixed strategy: more than one action is played with positive probability
        - these actions are called the support of the mixed strategy
- Let the set of all strategies for i be S_i
- Let the set of all strategy profiles be $S = S_1 \times \ldots \times S_n$.

Expected Utility and Best Response

Expected utility under a given mixed strategy profile $s \in S$:

$$u_i(s) = \sum_{a \in A} u_i(a) Pr(a|s)$$

$$Pr(a|s) = \prod_{j \in N} s_j(a_j)$$

Expected Utility and Best Response

Expected utility under a given mixed strategy profile $s \in S$:

$$u_i(s) = \sum_{a \in A} u_i(a) Pr(a|s)$$

$$Pr(a|s) = \prod_{j \in N} s_j(a_j)$$

- If you knew what everyone else was going to do, it would be easy to pick your own action
- Let $s_{-i} = \langle s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_n \rangle$; now $s = (s_{-i}, s_i)$

Definition (Best Response)

$s_i^* \in BR(s_{-i})$ iff $\forall s_i \in S_i$, $u_i(s_i^*, s_{-i}) \geq u_i(s_i, s_{-i})$.

Nash Equilibrium

- In general no agent knows what the others will do.
- What strategy profiles are "sensible"?

- Idea: look for stable strategy profiles.

Definition (Nash Equilibrium)

$s = \langle s_1, \ldots, s_n \rangle$ is a Nash equilibrium iff $\forall i,\ s_i \in BR(s_{-i})$.

Theorem (Nash, 1951)

Every finite game has at least one Nash equilibrium.

Why study equilibrium computation?

Because the concept of Nash equilibrium has proven important in many application areas.

- While it has limitations, Nash equilibrium is one of the key models of what behavior will emerge in noncooperative, multiagent interactions
- It is widely applied in economics, management science, operations research and finance, often with great success
  - recognized most prominently in Nash's Nobel prize
- Equilibrium and related concepts (e.g., ESS) are commonly used to study evolutionary biology and zoology
- It has also had substantial impact on government policy, and even on popular culture
  - For examples of the latter—and, to some extent, the former—Google "strangelove game theory" or "dark knight game theory"

Why study equilibrium computation?

...Because characterizing the complexity of equilibrium computation helps us to see how reasonable it is as a way of understanding games.

"If your laptop can't find the equilibrium, then neither can the market."
— Kamal Jain

Why study equilibrium computation?

...Because characterizing the complexity of equilibrium computation helps us to see how reasonable it is as a way of understanding games.

"If your laptop can't find the equilibrium, then neither can the market."
— Kamal Jain

...Because we need practical algorithms for computing equilibrium.

"[Due to the non-existence of efficient algorithms for computing equilibria], general equilibrium analysis has remained at a level of abstraction and mathematical theoretizing far removed from its ultimate purpose as a method for the evaluation of economic policy."
— Herbert Scarf (in his 1973 monograph on "The Computation of Economic Equilibria")

Beyond 2×2 Games

- When we use game theory to model real systems, we'd like to consider games with more than two agents and two actions
- Some examples of the kinds of questions we would like to be able to answer:
  - How will heterogeneous users route their traffic in a network?
  - How will advertisers bid in a sponsored search auction?
  - Which job skills will students choose to pursue?
  - Where in a city will businesses choose to locate?

Beyond 2×2 Games

- When we use game theory to model real systems, we'd like to consider games with more than two agents and two actions
- Some examples of the kinds of questions we would like to be able to answer:
  - How will heterogeneous users route their traffic in a network?
  - How will advertisers bid in a sponsored search auction?
  - Which job skills will students choose to pursue?
  - Where in a city will businesses choose to locate?
- Most GT work is analytic, not computational. What's holding us back?
  - a lack of game representations that can model interesting interactions in a reasonable amount of space
  - a lack of algorithms that can answer game-theoretic questions about these games in a reasonable amount of time
- In the past decade, substantial progress on both fronts

Overview

More Solution Concepts

- Solution concepts are rules that designate certain outcomes of a game as special or important
- We've already seen Nash equilibrium: strategy profiles in which all agents simultaneously best respond
- Nash equilibrium has advantages:
  - stability: given correct beliefs, no agent would change strategy
  - existence in all games

More Solution Concepts

- Solution concepts are rules that designate certain outcomes of a game as special or important
- We've already seen Nash equilibrium: strategy profiles in which all agents simultaneously best respond
- Nash equilibrium has advantages:
  - stability: given correct beliefs, no agent would change strategy
  - existence in all games
- It also has disadvantages:
  - may require agents to play mixed strategies
  - not prescriptive: only (necessarily) the right thing to do if other agents also play equilibrium strategies
  - doesn't account for stochastic information agents may share in common
  - assumes agents are perfect best responders
- Other solution concepts address these concerns...

Pure-Strategy Nash Equilibrium

What if we don't believe that agents would play mixed strategies?

Definition (Pure-Strategy Nash Equilibrium)

$a = \langle a_1, \ldots, a_n \rangle$ is a Pure-Strategy Nash equilibrium iff $\forall i,\, a_i \in BR(a_{-i})$.

- This is just like Nash equilibrium, but it requires all agents to play pure strategies
- Pure-strategy Nash equilibria are (arguably) more compelling than Nash equilibria, but not guaranteed to exist

Maxmin and Minmax

Definition (Maxmin)

In a two-player game, the maxmin strategy for player i is $\arg\max_{s_i} \min_{s_{-i}} u_i(s_1, s_2)$, and the maxmin value for player i is $\max_{s_i} \min_{s_{-i}} u_i(s_1, s_2)$.

- This is the most that agent i can guarantee himself, without making any assumptions about $-i$'s behavior.

Maxmin and Minmax

Definition (Maxmin)

In a two-player game, the maxmin strategy for player i is $\arg\max_{s_i} \min_{s_{-i}} u_i(s_1, s_2)$, and the maxmin value for player i is $\max_{s_i} \min_{s_{-i}} u_i(s_1, s_2)$.

- This is the most that agent i can guarantee himself, without making any assumptions about $-i$'s behavior.

Definition (Minmax)

In a two-player game, the minmax strategy for player i against player $-i$ is $\arg\min_{s_i} \max_{s_{-i}} u_{-i}(s_i, s_{-i})$, and player $-i$'s minmax value is $\min_{s_i} \max_{s_{-i}} u_{-i}(s_i, s_{-i})$.

- This is the least that agent i can guarantee that $-i$ will receive, ignoring his own payoffs.

A Special Case: Zero-Sum Games

In two-player zero-sum games, the Nash equilibrium has more prescriptive force than in the general case.

Theorem (Minimax theorem (von Neumann, 1928))

In any finite, two-player, zero-sum game, in any Nash equilibrium each player receives a payoff that is equal to both his maxmin value and his minmax value.

A Special Case: Zero-Sum Games

In two-player zero-sum games, the Nash equilibrium has more prescriptive force than in the general case.

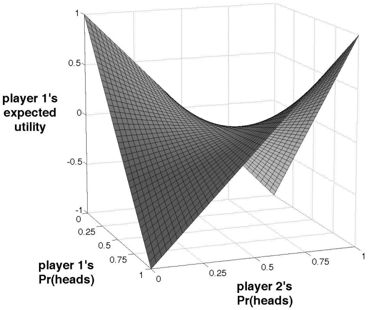
Theorem (Minimax theorem (von Neumann, 1928))

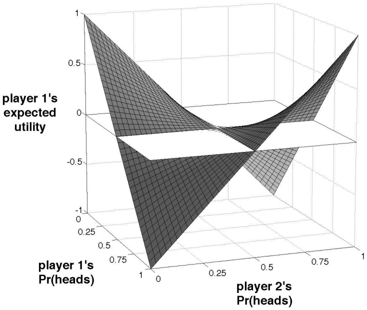
In any finite, two-player, zero-sum game, in any Nash equilibrium each player receives a payoff that is equal to both his maxmin value and his minmax value.

Consequences:

1. Each player's maxmin value is equal to his minmax value.
2. For both players, the set of maxmin strategies coincides with the set of minmax strategies.
3. Any maxmin strategy profile (or, equivalently, minmax strategy profile) is a Nash equilibrium. Furthermore, these are all the Nash equilibria. Thus, all Nash equilibria have the same payoff vector.

Saddle Point: Matching Pennies





Correlated Equilibrium

What if agents observe correlated random variables?

- Consider again Battle of the Sexes.
  - Intuitively, the best outcome seems a 50-50 split between (F, F) and (B, B).
  - But there's no way to achieve this, so either someone loses out (unfair) or both players often miscoordinate

Correlated Equilibrium

What if agents observe correlated random variables?

- Consider again Battle of the Sexes.
  - Intuitively, the best outcome seems a 50-50 split between (F, F) and (B, B).
  - But there's no way to achieve this, so either someone loses out (unfair) or both players often miscoordinate
- Another classic example: traffic game

	go	$wait$
go	$-100, -100$	$10, 0$
B	$0, 10$	$-10, -10$

- What is the natural solution here?

Correlated Equilibrium

What if agents observe correlated random variables?

- Consider again Battle of the Sexes.
  - Intuitively, the best outcome seems a 50-50 split between (F, F) and (B, B).
  - But there's no way to achieve this, so either someone loses out (unfair) or both players often miscoordinate
- Another classic example: traffic game

	go	$wait$
go	$-100, -100$	$10, 0$
B	$0, 10$	$-10, -10$

- What is the natural solution here?
  - A traffic light: fair randomizing devices that tell one of the agents to go and the other to wait.
  - the negative payoff outcomes are completely avoided
  - fairness is achieved
  - the sum of social welfare exceeds that of any Nash equilibrium

Correlated Equilibrium: Formal definition

Definition (Correlated equilibrium)

Given an n-agent game $G = (N, A, u)$, a correlated equilibrium is a tuple (v, π, σ), where v is a tuple of random variables $v = (v_1, \ldots, v_n)$ with respective domains $D = (D_1, \ldots, D_n)$, π is a joint distribution over v, $\sigma = (\sigma_1, \ldots, \sigma_n)$ is a vector of mappings $\sigma_i : D_i \mapsto A_i$, and for each agent i and every mapping $\sigma_i' : D_i \mapsto A_i$ it is the case that

$$\sum_{d \in D} \pi(d) u_i \left(\sigma_i(d_i), \sigma_{-i}(d_{-i})\right) \geq \sum_{d \in D} \pi(d) u_i \left(\sigma_i'(d_i), \sigma_{-i}(d_{-i})\right).$$

Theorem

For every Nash equilibrium σ^* there exists a corresponding correlated equilibrium σ. Thus, correlated equilibria always exist.

ϵ-Equilibrium

What if agents aren't perfect best responders?

Definition (ϵ-Nash, additive version)

Fix $\epsilon > 0$. A strategy profile s is an ϵ-Nash equilibrium (in the additive sense) if, for all agents i and for all strategies $s'_i \neq s_i$, $u_i(s_i, s_{-i}) \geq u_i(s'_i, s_{-i}) - \epsilon$.

Definition (ϵ-Nash, relative version)

Fix $\epsilon > 0$. A strategy profile s is an ϵ-Nash equilibrium (in the relative sense) if, for all agents i and for all strategies $s'_i \neq s_i$, $u_i(s_i, s_{-i}) \geq (1 - \epsilon) u_i(s'_i, s_{-i})$.

ϵ-Equilibrium

Advantages of these solution concepts:

- Every Nash equilibrium is surrounded by a region of ϵ-Nash equilibria for any $\epsilon > 0$.
- Seems convincing that agents should be indifferent to sufficiently small gains
- Methods for the "exact" computation of Nash equilibria that rely on floating point actually find only ϵ-equilibria (in the additive sense), where ϵ is roughly 10^{-16}.

ϵ-Equilibrium

Drawbacks of these solution concepts (both variants):

- ϵ-Nash equilibria are not necessarily close to any Nash equilibrium.
  - This undermines the sense in which ϵ-Nash equilibria can be understood as approximations of Nash equilibria.
- ϵ-Nash equilibria can have payoffs arbitrarily lower than those of any Nash equilibrium
- ϵ-Nash equilibria can even involve dominated strategies.

Overview

1. Plan of this Tutorial
2. Getting Our Bearings: A Quick Game Theory Refresher
3. Solution Concepts
4. Computational Formulations

Computing Mixed Nash Equilibria: Battle of the Sexes

	B	F
B	$2, 1$	$0, 0$
F	$0, 0$	$1, 2$

- For Battle of the Sexes, let's look for an equilibrium where all actions are part of the support

Computing Mixed Nash Equilibria: Battle of the Sexes

	B	F
B	$2, 1$	$0, 0$
F	$0, 0$	$1, 2$

- Let player 2 play B with p, F with $1 - p$.
- If player 1 best-responds with a mixed strategy, player 2 must make him indifferent between F and B

$$\begin{aligned} u_1(B) &= u_1(F) \\ 2p + 0(1-p) &= 0p + 1(1-p) \\ p &= \frac{1}{3} \end{aligned}$$

Computing Mixed Nash Equilibria: Battle of the Sexes

	B	F
B	$2, 1$	$0, 0$
F	$0, 0$	$1, 2$

- Likewise, player 1 must randomize to make player 2 indifferent.
- Let player 1 play B with q, F with $1 - q$.

$$
\begin{aligned}
u_2(B) &= u_2(F) \\
q + 0(1-q) &= 0q + 2(1-q) \\
q &= \frac{2}{3}
\end{aligned}
$$

- Thus the strategies $(\frac{2}{3}, \frac{1}{3})$, $(\frac{1}{3}, \frac{2}{3})$ are a Nash equilibrium.

Computing Mixed Nash Equilibria: Battle of the Sexes

Advantages of this approach:

- At least for a 2×2 game, this was computationally feasible
  - in general, when checking non-full supports, it's a linear program, because we have to ensure that actions outside the support aren't better

Computing Mixed Nash Equilibria: Battle of the Sexes

Advantages of this approach:

- At least for a 2×2 game, this was computationally feasible
  - in general, when checking non-full supports, it's a linear program, because we have to ensure that actions outside the support aren't better

Disadvantages of this approach:

- We had to start by correctly guessing the support
- There are $\prod_{i \in N} 2^{|A_i|}$ supports that we'd have to check

Computing Mixed Nash Equilibria: Battle of the Sexes

Advantages of this approach:

- At least for a 2×2 game, this was computationally feasible
  - in general, when checking non-full supports, it's a linear program, because we have to ensure that actions outside the support aren't better

Disadvantages of this approach:

- We had to start by correctly guessing the support
- There are $\prod_{i \in N} 2^{|A_i|}$ supports that we'd have to check

This method is going to have pretty awful worst-case performance as games get much larger than 2×2.[1]

---

[1] Interesting caveat: in fact, if combined with the right heuristics, support enumeration can be a competitive approach for finding equilibria. See [Porter, Nudelman & Shoham, 2004].

Computational Formulations

- Now we'll look at the computational problems of identifying
  - pure-strategy Nash equilibria
  - correlated equilibria
  - Nash equilibria of two-player, zero-sum games
- In each case, we'll consider how the problem differs from that of computing NE of general-sum games (NASH)
- Ultimately, we aim to illustrate why the NASH problem is so different from these other problems, and why its complexity was so tricky to characterize.

Computing Pure-Strategy Nash Equilibrium

Constraint Satisfaction Problem

Find $a \in A$ such that $\forall i$, $a_i \in BR(a_{-i})$.

Computing Pure-Strategy Nash Equilibrium

Constraint Satisfaction Problem

Find $a \in A$ such that $\forall i$, $a_i \in BR(a_{-i})$.

- This is an easy problem to solve:
  - note that the input size is $O(n|A|)$
  - checking whether a given $a \in A$ involves a BR for player i requires $O(|A_i|)$ time, which is $O(|A|)$
  - there are $|A|$ strategy profiles to check
  - thus, we can solve the problem in $O(|A|^2)$ time

Computing Pure-Strategy Nash Equilibrium

Constraint Satisfaction Problem

Find $a \in A$ such that $\forall i$, $a_i \in BR(a_{-i})$.

- This is an easy problem to solve:
  - note that the input size is $O(n|A|)$
  - checking whether a given $a \in A$ involves a BR for player i requires $O(|A_i|)$ time, which is $O(|A|)$
  - there are $|A|$ strategy profiles to check
  - thus, we can solve the problem in $O(|A|^2)$ time
- However, we won't be able to find (general) Nash equilibria by enumerating them
  - Thus, this result seems unlikely to carry over straightforwardly...

Computing Correlated Equilibrium

Linear Feasibility Program

$$\sum_{a \in A | a_i \in a} p(a) u_i(a) \geq \sum_{a \in A | a_i \in a} p(a) u_i(a'_i, a_{-i}) \quad \forall i \in N,\ \forall a_i, a'_i \in A_i$$

$$p(a) \geq 0 \quad \forall a \in A$$

$$\sum_{a \in A} p(a) = 1$$

- variables: $p(a)$; constants: $u_i(a)$

Computing Correlated Equilibrium

Linear Feasibility Program

$$\sum_{a \in A | a_i \in a} p(a) u_i(a) \geq \sum_{a \in A | a_i \in a} p(a) u_i(a_i', a_{-i}) \quad \forall i \in N, \forall a_i, a_i' \in A_i$$

$$p(a) \geq 0 \quad \forall a \in A$$

$$\sum_{a \in A} p(a) = 1$$

- variables: $p(a)$; constants: $u_i(a)$
- we could find the social-welfare maximizing CE by adding an objective function

$$\text{maximize:} \quad \sum_{a \in A} p(a) \sum_{i \in N} u_i(a).$$

Computing Correlated Equilibrium

Linear Feasibility Program

$$\sum_{a \in A | a_i \in a} p(a) u_i(a) \geq \sum_{a \in A | a'_i \in a} p(a) u_i(a'_i, a_{-i}) \quad \forall i \in N, \forall a_i, a'_i \in A_i$$

$$p(a) \geq 0 \quad \forall a \in A$$

$$\sum_{a \in A} p(a) = 1$$

Why can't we compute NE like we did CE?

- intuitively, correlated equilibrium has only a single randomization over outcomes, whereas in NE this is constructed as a product of independent probabilities.
- To find NE, the first constraint would have to be nonlinear:

$$\sum_{a \in A} u_i(a) \prod_{j \in N} p_j(a_j) \geq \sum_{a \in A} u_i(a'_i, a_{-i}) \prod_{j \in N \setminus \{i\}} p_j(a_j) \quad \forall i \in N, \forall a'_i \in A_i.$$

Computing Equilibria of Zero-Sum Games

Linear Program

$$\begin{aligned} \text{minimize} \quad & U_1^* \\ \text{subject to} \quad & \sum_{a_2 \in A_2} u_1(a_1, a_2) \cdot s_2^{a_2} \leq U_1^* && \forall a_1 \in A_1 \\ & \sum_{a_2 \in A_2} s_2^{a_2} = 1 \\ & s_2^{a_2} \geq 0 && \forall a_2 \in A_2 \end{aligned}$$

- First, identify the variables:
  - U_1^* is the expected utility for player 1
  - $s_2^{a_2}$ is player 2's probability of playing action a_2 under his mixed strategy
- each $u_1(a_1, a_2)$ is a constant.

Computing Equilibria of Zero-Sum Games

Now let's interpret the LP:

Linear Program

$$
\begin{aligned}
\text{minimize} \quad & U_1^* \\
\text{subject to} \quad & \sum_{a_2 \in A_2} u_1(a_1, a_2) \cdot s_2^{a_2} \leq U_1^* \qquad \forall a_1 \in A_1 \\
& \sum_{a_2 \in A_2} s_2^{a_2} = 1 \\
& s_2^{a_2} \geq 0 \qquad \forall a_2 \in A_2
\end{aligned}
$$

- s_2 is a valid probability distribution.

Computing Equilibria of Zero-Sum Games

Now let's interpret the LP:

Linear Program

$$
\begin{aligned}
\textcolor{red}{\text{minimize}} \quad & \textcolor{red}{U_1^*} \\
\text{subject to} \quad & \sum_{a_2 \in A_2} u_1(a_1, a_2) \cdot s_2^{a_2} \leq U_1^* && \forall a_1 \in A_1 \\
& \sum_{a_2 \in A_2} s_2^{a_2} = 1 \\
& s_2^{a_2} \geq 0 && \forall a_2 \in A_2
\end{aligned}
$$

- U_1^* is as small as possible.

Computing Equilibria of Zero-Sum Games

Now let's interpret the LP:

Linear Program

$$
\begin{aligned}
\text{minimize} \quad & U_1^* \\
\text{subject to} \quad & \sum_{a_2 \in A_2} u_1(a_1, a_2) \cdot s_2^{a_2} \leq U_1^* && \forall a_1 \in A_1 \\
& \sum_{a_2 \in A_2} s_2^{a_2} = 1 \\
& s_2^{a_2} \geq 0 && \forall a_2 \in A_2
\end{aligned}
$$

- Player 1's expected utility for playing each of his actions under player 2's mixed strategy is no more than U_1^*.
  - Because U_1^* is minimized, this constraint will be tight for some actions: the support of player 1's mixed strategy.

Computing Equilibria of Zero-Sum Games

Linear Program

$$
\begin{aligned}
\text{minimize} \quad & U_1^* \\
\text{subject to} \quad & \sum_{a_2 \in A_2} u_1(a_1, a_2) \cdot s_2^{a_2} \leq U_1^* && \forall a_1 \in A_1 \\
& \sum_{a_2 \in A_2} s_2^{a_2} = 1 \\
& s_2^{a_2} \geq 0 && \forall a_2 \in A_2
\end{aligned}
$$

- This formulation gives us the minmax strategy for player 2.
- To get the minmax strategy for player 1, we need to solve a second (analogous) LP.

Computing Equilibria of Zero-Sum Games

We can reformulate the LP using slack variables, as follows:

Linear Program

$$\begin{aligned}
\text{minimize} \quad & U_1^* \\
\text{subject to} \quad & \sum_{a_2 \in A_2} u_1(a_1, a_2) \cdot s_2^{a_2} + r_1^{a_1} = U_1^* && \forall a_1 \in A_1 \\
& \sum_{a_2 \in A_2} s_2^{a_2} = 1 \\
& s_2^{a_2} \geq 0 && \forall a_2 \in A_2 \\
& r_1^{a_1} \geq 0 && \forall a_1 \in A_1
\end{aligned}$$

All we've done is change the weak inequality into an equality by adding a nonnegative variable.

Computing Nash Equilibria of General, Two-Player Games

We can generalize the previous LP to derive a formulation for computing a NE of a general-sum, two-player game.

Linear Complementarity Problem

$$\sum_{a_2 \in A_2} u_1(a_1, a_2) \cdot s_2^{a_2} + r_1^{a_1} = U_1^* \qquad \forall a_1 \in A_1$$

$$\sum_{a_1 \in A_1} u_2(a_1, a_2) \cdot s_1^{a_1} + r_2^{a_2} = U_2^* \qquad \forall a_2 \in A_2$$

$$\sum_{a_1 \in A_1} s_1^{a_1} = 1, \quad \sum_{a_2 \in A_2} s_2^{a_2} = 1$$

$$s_1^{a_1} \geq 0, \quad s_2^{a_2} \geq 0 \qquad \forall a_1 \in A_1, \forall a_2 \in A_2$$

$$r_1^{a_1} \geq 0, \quad r_2^{a_2} \geq 0 \qquad \forall a_1 \in A_1, \forall a_2 \in A_2$$

$$r_1^{a_1} \cdot s_1^{a_1} = 0, \quad r_2^{a_2} \cdot s_2^{a_2} = 0 \qquad \forall a_1 \in A_1, \forall a_2 \in A_2$$

Note a strong resemblance to the previous LP with slack variables, but the absence of an objective function.

Computing Nash Equilibria of General, Two-Player Games

Linear Complementarity Problem

$$\sum_{a_2 \in A_2} u_1(a_1, a_2) \cdot s_2^{a_2} + r_1^{a_1} = U_1^* \qquad \forall a_1 \in A_1$$

$$\sum_{a_1 \in A_1} u_2(a_1, a_2) \cdot s_1^{a_1} + r_2^{a_2} = U_2^* \qquad \forall a_2 \in A_2$$

$$\sum_{a_1 \in A_1} s_1^{a_1} = 1, \quad \sum_{a_2 \in A_2} s_2^{a_2} = 1$$

$$s_1^{a_1} \geq 0, \quad s_2^{a_2} \geq 0 \qquad \forall a_1 \in A_1, \forall a_2 \in A_2$$

$$r_1^{a_1} \geq 0, \quad r_2^{a_2} \geq 0 \qquad \forall a_1 \in A_1, \forall a_2 \in A_2$$

$$r_1^{a_1} \cdot s_1^{a_1} = 0, \quad r_2^{a_2} \cdot s_2^{a_2} = 0 \qquad \forall a_1 \in A_1, \forall a_2 \in A_2$$

These are the same constraints as before.

Computing Nash Equilibria of General, Two-Player Games

Linear Complementarity Problem

$$\sum_{a_2 \in A_2} u_1(a_1, a_2) \cdot s_2^{a_2} + r_1^{a_1} = U_1^* \qquad \forall a_1 \in A_1$$

$$\sum_{a_1 \in A_1} u_2(a_1, a_2) \cdot s_1^{a_1} + r_2^{a_2} = U_2^* \qquad \forall a_2 \in A_2$$

$$\sum_{a_1 \in A_1} s_1^{a_1} = 1, \quad \sum_{a_2 \in A_2} s_2^{a_2} = 1$$

$$s_1^{a_1} \geq 0, \quad s_2^{a_2} \geq 0 \qquad \forall a_1 \in A_1, \forall a_2 \in A_2$$

$$r_1^{a_1} \geq 0, \quad r_2^{a_2} \geq 0 \qquad \forall a_1 \in A_1, \forall a_2 \in A_2$$

$$r_1^{a_1} \cdot s_1^{a_1} = 0, \quad r_2^{a_2} \cdot s_2^{a_2} = 0 \qquad \forall a_1 \in A_1, \forall a_2 \in A_2$$

Now we also add corresponding constraints for player 2.

Computing Nash Equilibria of General, Two-Player Games

Linear Complementarity Problem

$$\sum_{a_2 \in A_2} u_1(a_1, a_2) \cdot s_2^{a_2} + r_1^{a_1} = U_1^* \qquad \forall a_1 \in A_1$$

$$\sum_{a_1 \in A_1} u_2(a_1, a_2) \cdot s_1^{a_1} + r_2^{a_2} = U_2^* \qquad \forall a_2 \in A_2$$

$$\sum_{a_1 \in A_1} s_1^{a_1} = 1, \quad \sum_{a_2 \in A_2} s_2^{a_2} = 1$$

$$s_1^{a_1} \geq 0, \quad s_2^{a_2} \geq 0 \qquad \forall a_1 \in A_1, \forall a_2 \in A_2$$

$$r_1^{a_1} \geq 0, \quad r_2^{a_2} \geq 0 \qquad \forall a_1 \in A_1, \forall a_2 \in A_2$$

$$r_1^{a_1} \cdot s_1^{a_1} = 0, \quad r_2^{a_2} \cdot s_2^{a_2} = 0 \qquad \forall a_1 \in A_1, \forall a_2 \in A_2$$

Standard constraints on probabilities and slack variables.

Computing Nash Equilibria of General, Two-Player Games

Linear Complementarity Problem

$$\sum_{a_2 \in A_2} u_1(a_1, a_2) \cdot s_2^{a_2} + r_1^{a_1} = U_1^* \qquad \forall a_1 \in A_1$$

$$\sum_{a_1 \in A_1} u_2(a_1, a_2) \cdot s_1^{a_1} + r_2^{a_2} = U_2^* \qquad \forall a_2 \in A_2$$

$$\sum_{a_1 \in A_1} s_1^{a_1} = 1, \quad \sum_{a_2 \in A_2} s_2^{a_2} = 1$$

$$s_1^{a_1} \geq 0, \quad s_2^{a_2} \geq 0 \qquad \forall a_1 \in A_1, \forall a_2 \in A_2$$

$$r_1^{a_1} \geq 0, \quad r_2^{a_2} \geq 0 \qquad \forall a_1 \in A_1, \forall a_2 \in A_2$$

$$r_1^{a_1} \cdot s_1^{a_1} = 0, \quad r_2^{a_2} \cdot s_2^{a_2} = 0 \qquad \forall a_1 \in A_1, \forall a_2 \in A_2$$

With all of this, we'd have an LP, but the slack variables—and hence U_1^* and U_2^*—would be allowed to take unboundedly large values.

Computing Nash Equilibria of General, Two-Player Games

Linear Complementarity Problem

$$\sum_{a_2 \in A_2} u_1(a_1, a_2) \cdot s_2^{a_2} + r_1^{a_1} = U_1^* \qquad \forall a_1 \in A_1$$

$$\sum_{a_1 \in A_1} u_2(a_1, a_2) \cdot s_1^{a_1} + r_2^{a_2} = U_2^* \qquad \forall a_2 \in A_2$$

$$\sum_{a_1 \in A_1} s_1^{a_1} = 1, \quad \sum_{a_2 \in A_2} s_2^{a_2} = 1$$

$$s_1^{a_1} \geq 0, \quad s_2^{a_2} \geq 0 \qquad \forall a_1 \in A_1, \forall a_2 \in A_2$$

$$r_1^{a_1} \geq 0, \quad r_2^{a_2} \geq 0 \qquad \forall a_1 \in A_1, \forall a_2 \in A_2$$

$$r_1^{a_1} \cdot s_1^{a_1} = 0, \quad r_2^{a_2} \cdot s_2^{a_2} = 0 \qquad \forall a_1 \in A_1, \forall a_2 \in A_2$$

Complementary slackness condition: whenever an action is in the support of a given player's mixed strategy then the corresponding slack variable must be zero (i.e., the constraint must be tight).

Computing Nash Equilibria of General, Two-Player Games

Linear Complementarity Problem

$$\sum_{a_2 \in A_2} u_1(a_1, a_2) \cdot s_2^{a_2} + r_1^{a_1} = U_1^* \qquad \forall a_1 \in A_1$$

$$\sum_{a_1 \in A_1} u_2(a_1, a_2) \cdot s_1^{a_1} + r_2^{a_2} = U_2^* \qquad \forall a_2 \in A_2$$

$$\sum_{a_1 \in A_1} s_1^{a_1} = 1, \quad \sum_{a_2 \in A_2} s_2^{a_2} = 1$$

$$s_1^{a_1} \geq 0, \quad s_2^{a_2} \geq 0 \qquad \forall a_1 \in A_1, \forall a_2 \in A_2$$

$$r_1^{a_1} \geq 0, \quad r_2^{a_2} \geq 0 \qquad \forall a_1 \in A_1, \forall a_2 \in A_2$$

$$r_1^{a_1} \cdot s_1^{a_1} = 0, \quad r_2^{a_2} \cdot s_2^{a_2} = 0 \qquad \forall a_1 \in A_1, \forall a_2 \in A_2$$

Each slack variable can be viewed as the player's incentive to deviate from the corresponding action. Thus, in equilibrium, all strategies that are played with positive probability must yield the same expected payoff, while all strategies that lead to lower expected payoffs are not played.

Computing Nash Equilibria of General, Two-Player Games

Linear Complementarity Problem

$$\sum_{a_2 \in A_2} u_1(a_1, a_2) \cdot s_2^{a_2} + r_1^{a_1} = U_1^* \qquad \forall a_1 \in A_1$$

$$\sum_{a_1 \in A_1} u_2(a_1, a_2) \cdot s_1^{a_1} + r_2^{a_2} = U_2^* \qquad \forall a_2 \in A_2$$

$$\sum_{a_1 \in A_1} s_1^{a_1} = 1, \quad \sum_{a_2 \in A_2} s_2^{a_2} = 1$$

$$s_1^{a_1} \geq 0, \quad s_2^{a_2} \geq 0 \qquad \forall a_1 \in A_1, \forall a_2 \in A_2$$

$$r_1^{a_1} \geq 0, \quad r_2^{a_2} \geq 0 \qquad \forall a_1 \in A_1, \forall a_2 \in A_2$$

$$r_1^{a_1} \cdot s_1^{a_1} = 0, \quad r_2^{a_2} \cdot s_2^{a_2} = 0 \qquad \forall a_1 \in A_1, \forall a_2 \in A_2$$

We are left with the requirement that each player plays a best response to the other player's mixed strategy: the definition of a Nash equilibrium.

Computing Nash Equilibria of General, Two-Player Games

Linear Complementarity Problem

$$\sum_{a_2 \in A_2} u_1(a_1, a_2) \cdot s_2^{a_2} + r_1^{a_1} = U_1^* \qquad \forall a_1 \in A_1$$

$$\sum_{a_1 \in A_1} u_2(a_1, a_2) \cdot s_1^{a_1} + r_2^{a_2} = U_2^* \qquad \forall a_2 \in A_2$$

$$\sum_{a_1 \in A_1} s_1^{a_1} = 1, \quad \sum_{a_2 \in A_2} s_2^{a_2} = 1$$

$$s_1^{a_1} \geq 0, \quad s_2^{a_2} \geq 0 \qquad \forall a_1 \in A_1, \forall a_2 \in A_2$$

$$r_1^{a_1} \geq 0, \quad r_2^{a_2} \geq 0 \qquad \forall a_1 \in A_1, \forall a_2 \in A_2$$

$$r_1^{a_1} \cdot s_1^{a_1} = 0, \quad r_2^{a_2} \cdot s_2^{a_2} = 0 \qquad \forall a_1 \in A_1, \forall a_2 \in A_2$$

Unfortunately, this LCP formulation doesn't imply polynomial time complexity the way an LP formulation does.

- However, it will be useful in what follows.

Complexity of NASH

We've seen how to compute:

- Pure-strategy Nash equilibria
- Correlated equilibria
- Equilibria of zero-sum, two-player games

In each case, we've seen evidence that the NASH problem is fundamentally different, even in its two-player variant.

Now Costis will take over, and investigate this question in more detail...

Equilibrium Computation in Normal Form Games

Costis Daskalakis & Kevin Leyton-Brown

Part 1(b)

Overview

- A brief history of the Nash Equilibrium.

- The complexity landscape between P and NP.

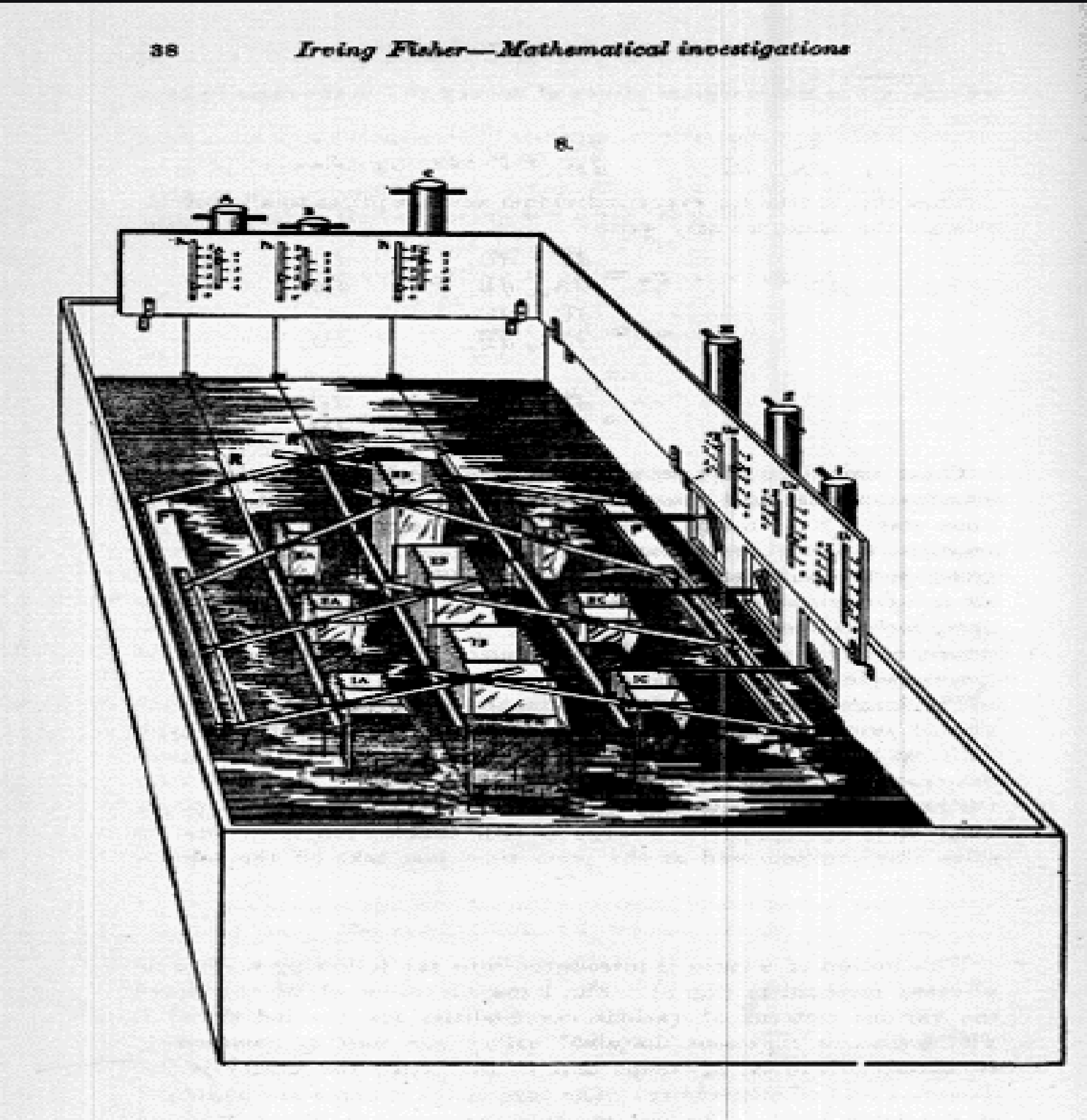
- The Complexity of the Nash Equilibrium.

The first computational thoughts

1891 Irving Fisher:

- Hydraulic apparatus for calculating the equilibrium of a related, *market model.*

- No existence proof for the general setting; but the machine would work for *3 traders and 3 commodities.*



History (cont.)

1928 Neumann: existence of Equilibrium in *2-player*, *zero-sum* games

proof uses Brouwer's fixed point theorem;

+ Danzig '57: equivalent to LP duality;

+ Khachiyan'79: polynomial-time solvable.

1950 Nash: existence of Equilibrium in *multiplayer*, *general-sum* games

proof also uses Brouwer's fixed point theorem;

intense effort for equilibrium algorithms:

Kuhn '61, Mangasarian '64, Lemke-Howson '64,
Rosenmüller '71, Wilson '71, Scarf '67, Eaves '72,
Laan-Talman '79, and others…

Lemke-Howson: simplex-like, works with LCP formulation;
no efficient algorithm is known after 50+ years of research.

the Pavlovian reaction

"Is it NP-complete to find a Nash equilibrium?"

two answers

1. probably not, since a solution is guaranteed to exist…

2. it is NP-complete to find a "tiny" bit more info than "just" a Nash equilibrium; e.g., the following are NP-complete:

  - find two Nash equilibria, if more than one exist
  - find a Nash equilibrium whose third bit is one, if any

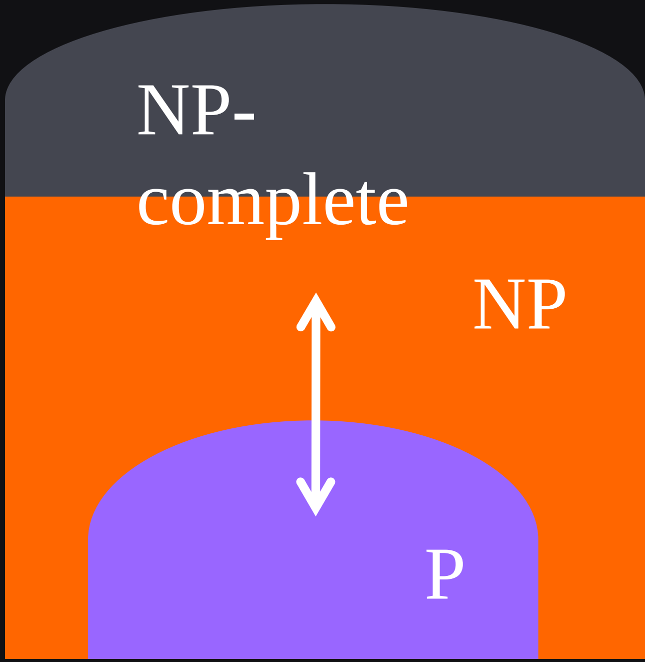
[Gilboa, Zemel '89; Conitzer, Sandholm '03]

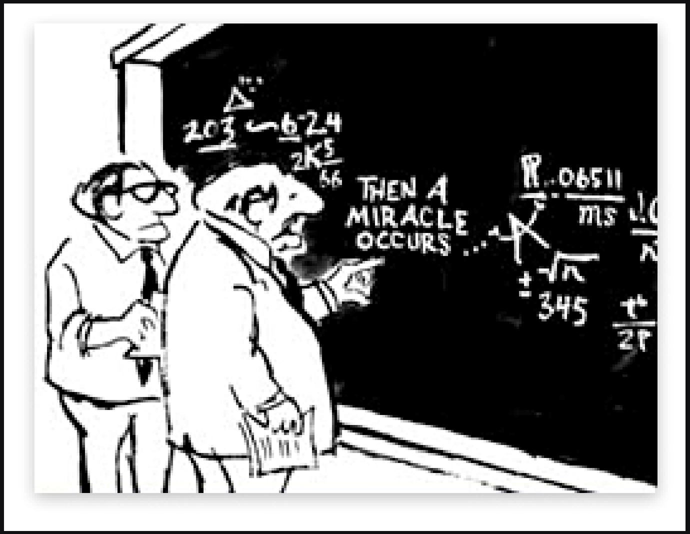
what about a single equilibrium?

- the theory of NP-completeness does not seem appropriate;

- in fact, NASH seems to lie below NP;

- making Nash's theorem constructive…

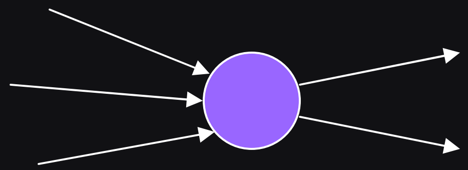




The Non-Constructive Step

an easy parity lemma:

a directed graph with an unbalanced node (a node with indegree ≠ outdegree) must have another.

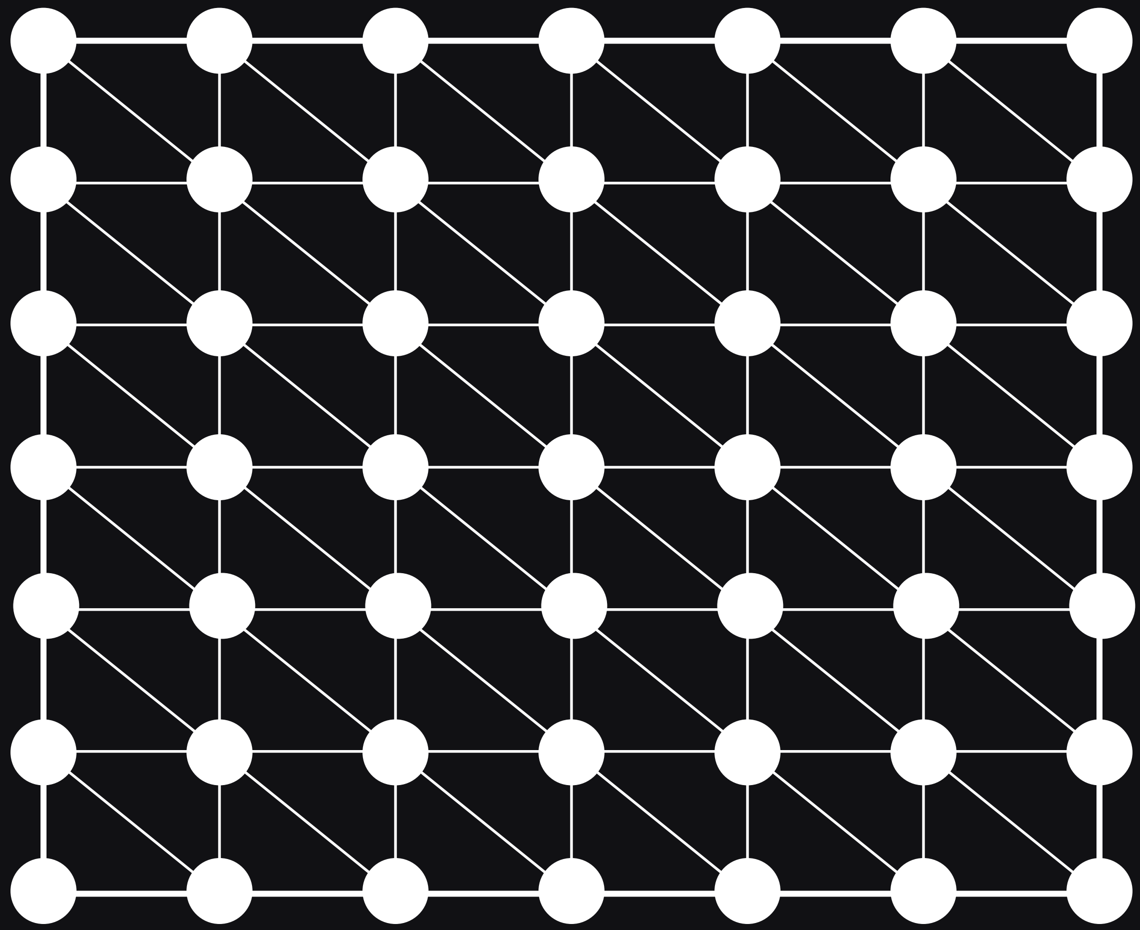


but, why is this non-constructive?

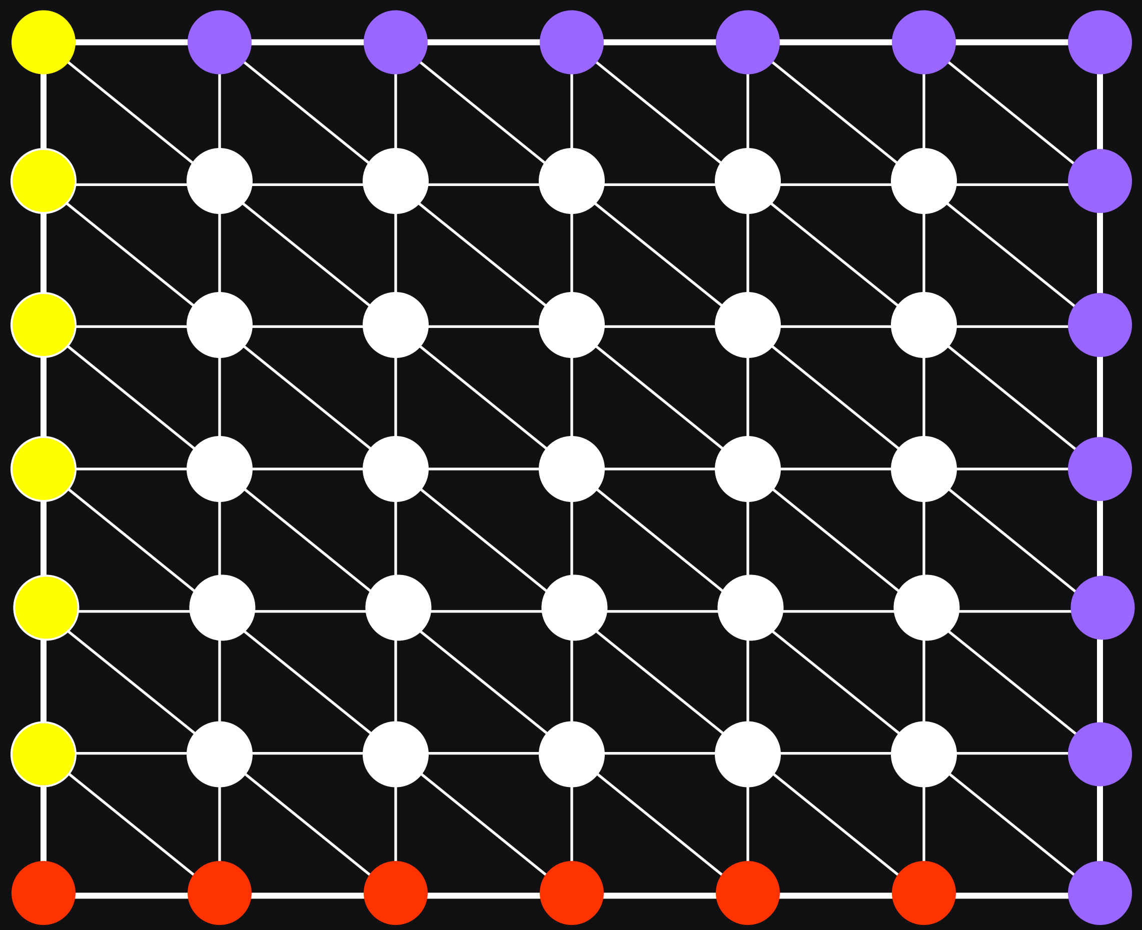
given a directed graph and an unbalanced node, isn't it trivial to find another unbalanced node?

the graph may be exponentially large, but have a succinct description… (more on this soon)

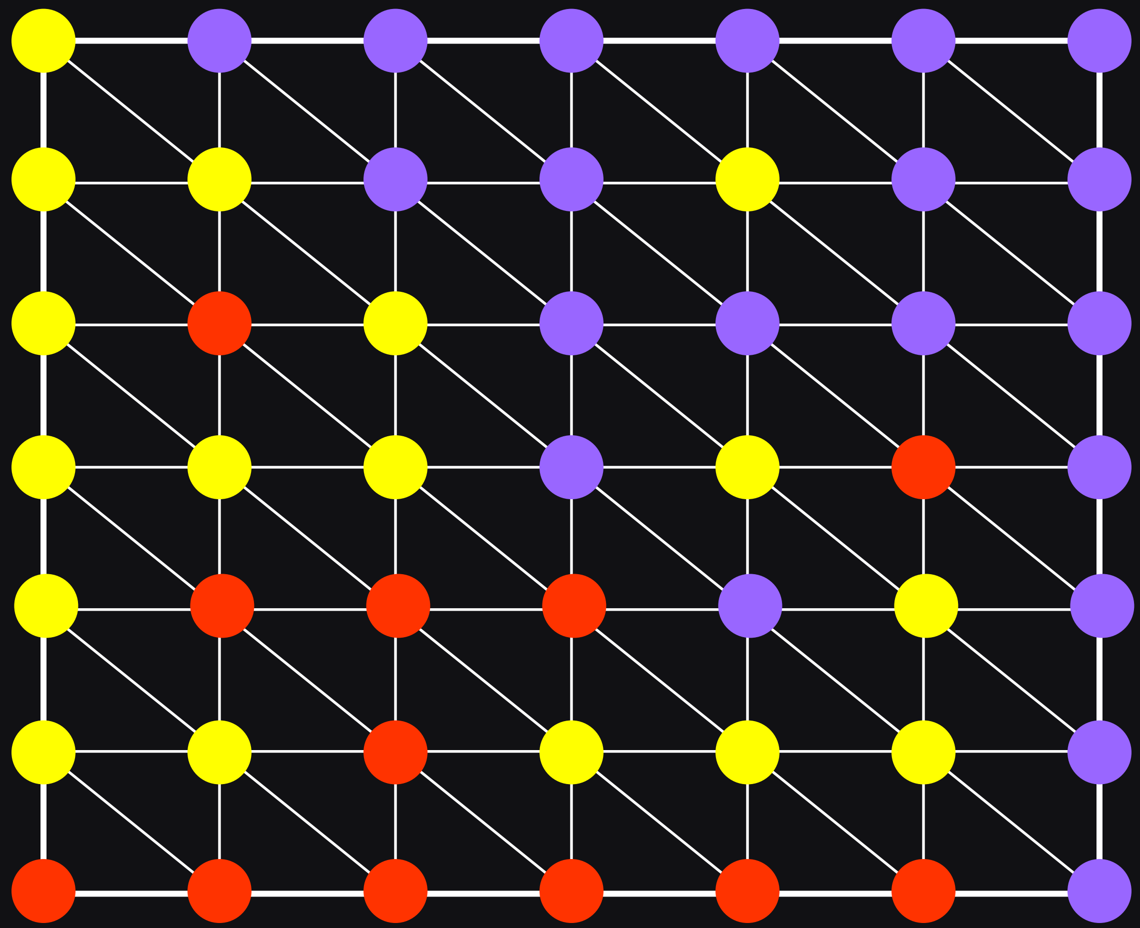
Sperner's Lemma



Sperner's Lemma

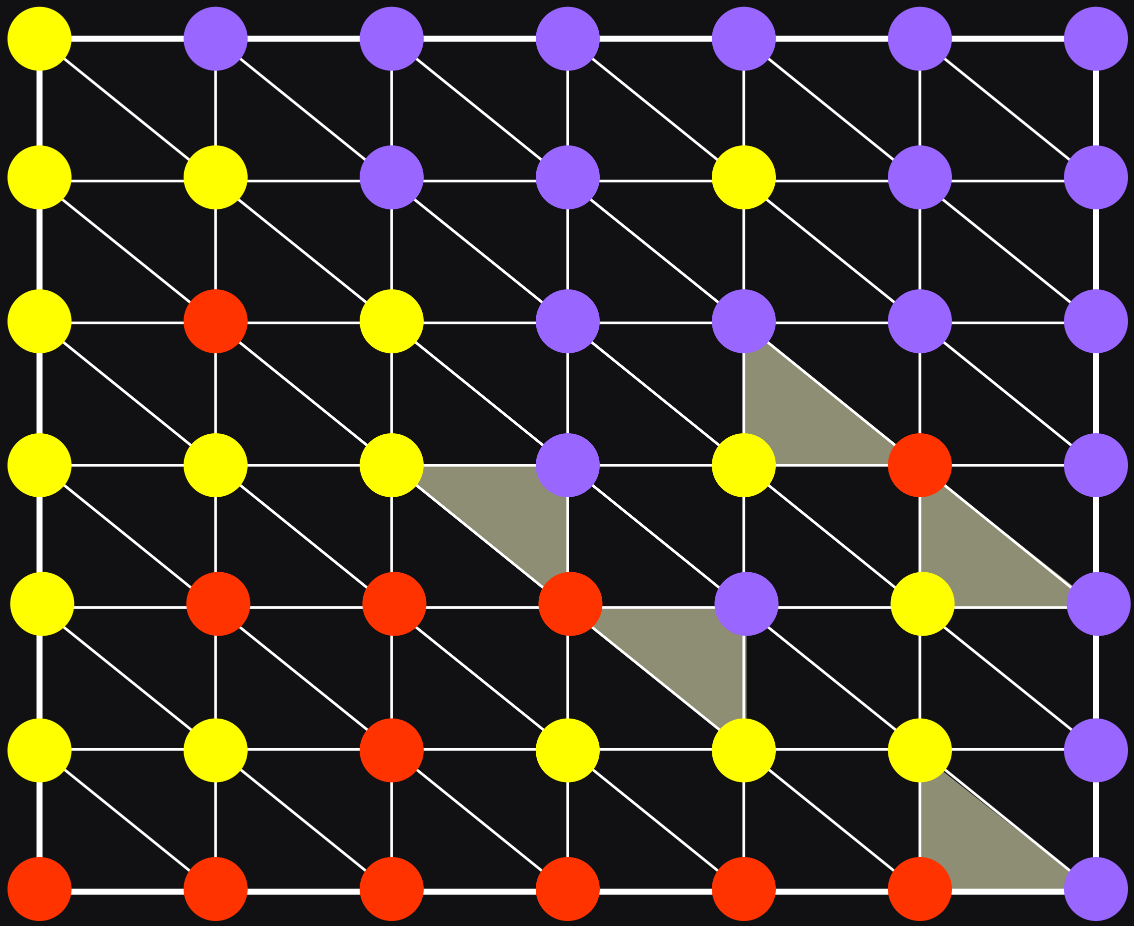


Sperner's Lemma



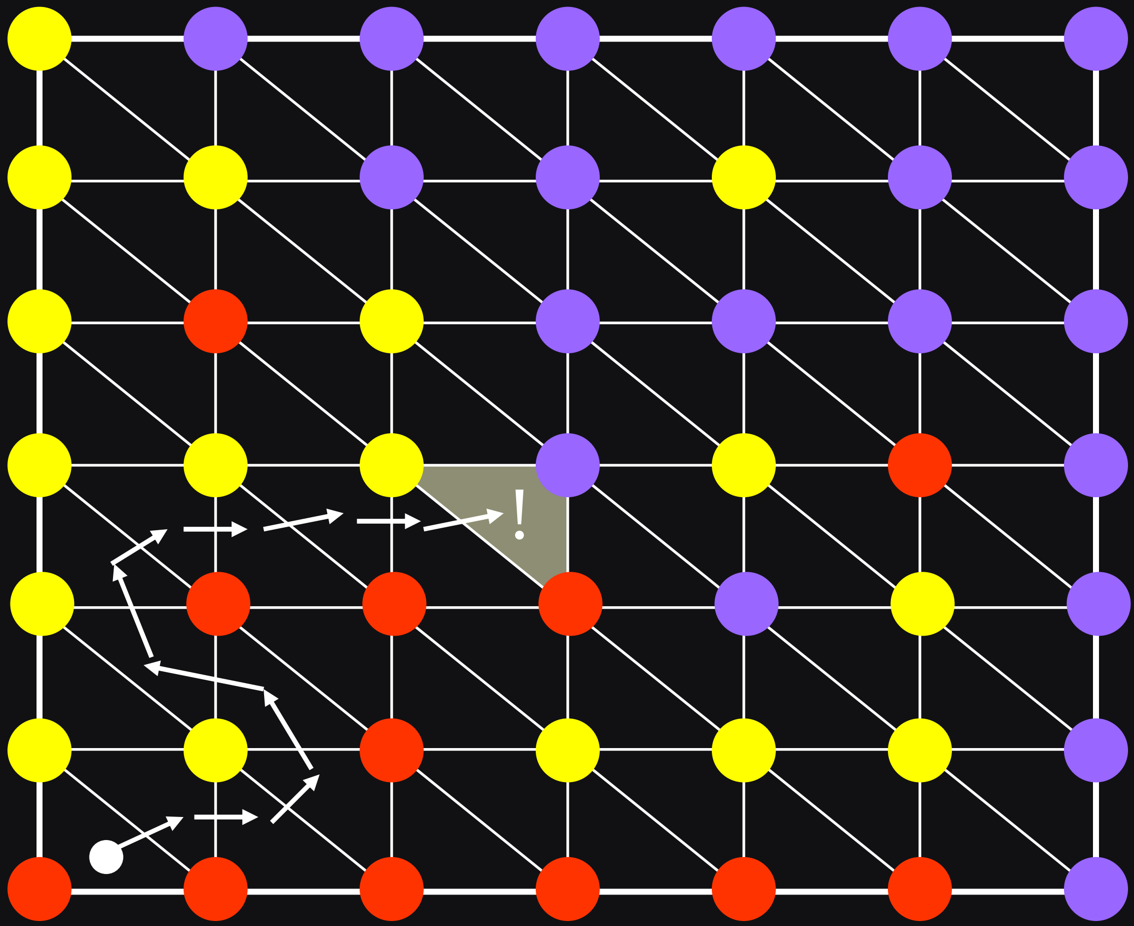
Lemma: No matter how the internal nodes are colored there exists a tri-chromatic triangle. In fact, an odd number of them.

Sperner's Lemma



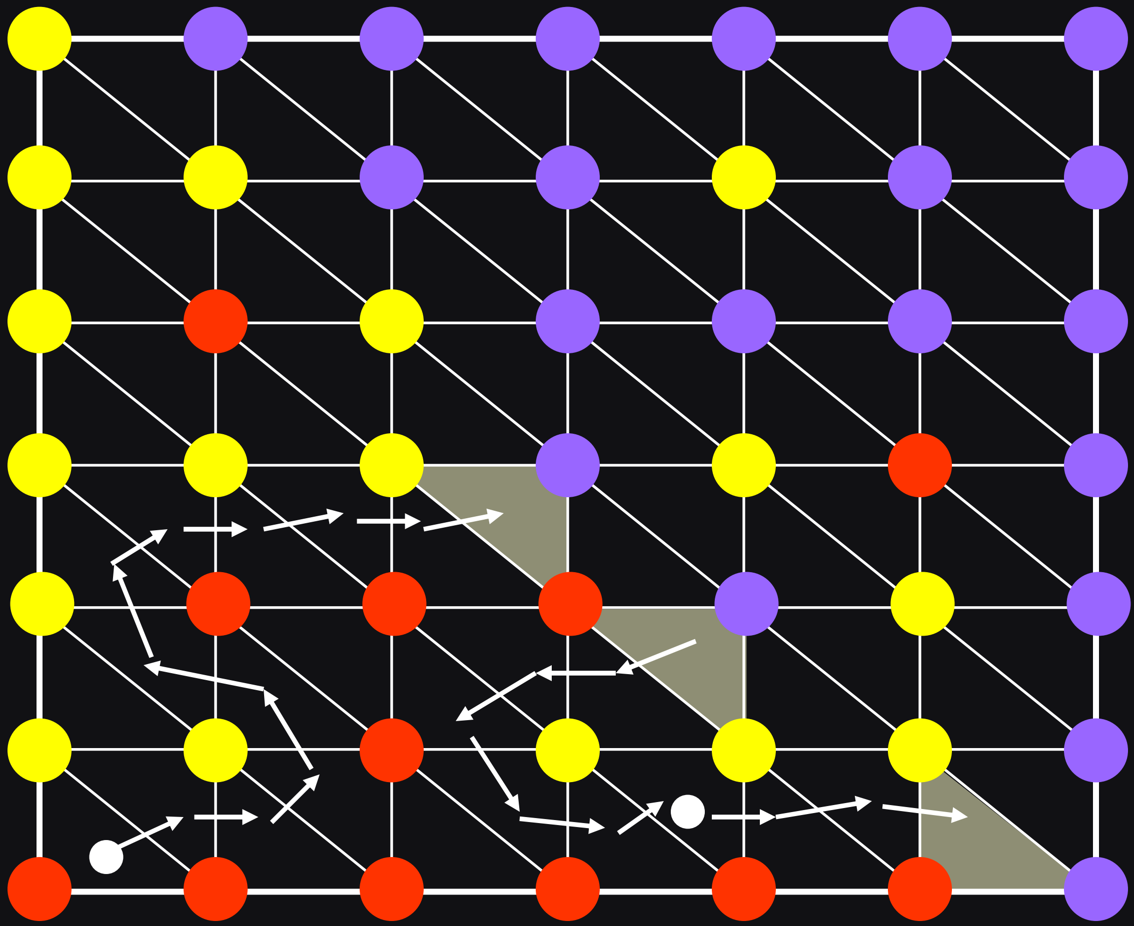
Lemma: No matter how the internal nodes are colored there exists a tri-chromatic triangle. In fact, an odd number of them.

Sperner's Lemma



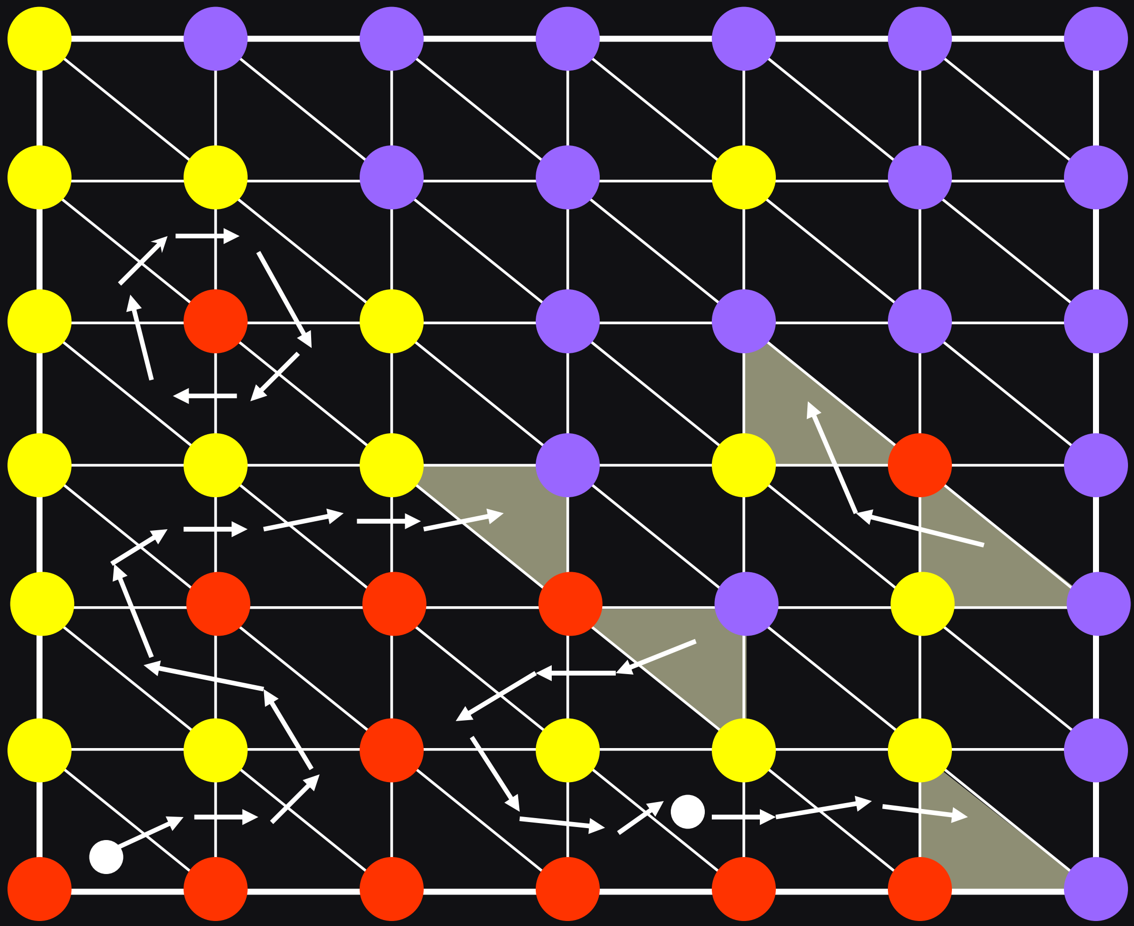
Lemma: No matter how the internal nodes are colored there exists a tri-chromatic triangle. In fact, an odd number of them.

Sperner's Lemma



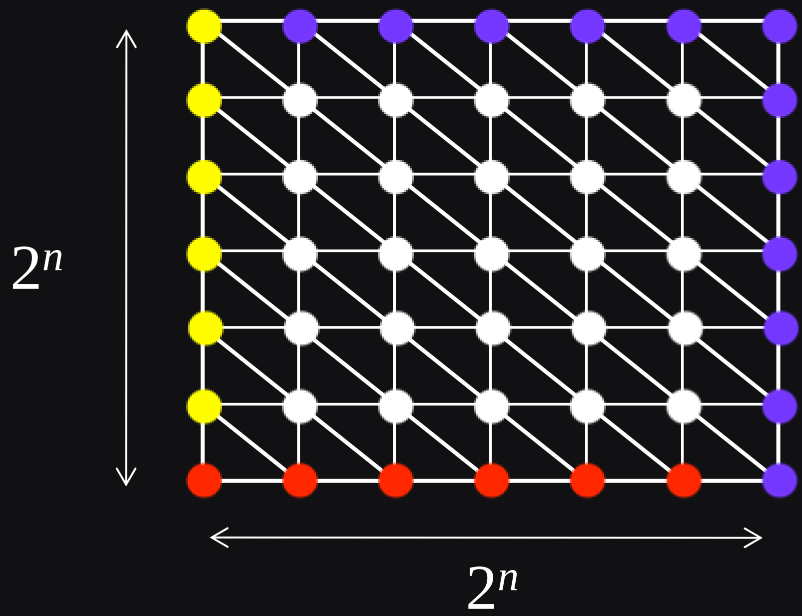
Lemma: No matter how the internal nodes are colored there exists a tri-chromatic triangle. In fact, an odd number of them.

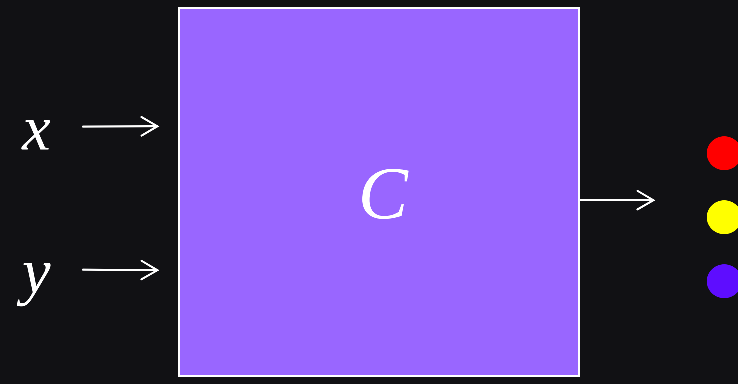
Sperner's Lemma



Lemma: No matter how the internal nodes are colored there exists a tri-chromatic triangle. In fact, an odd number of them.

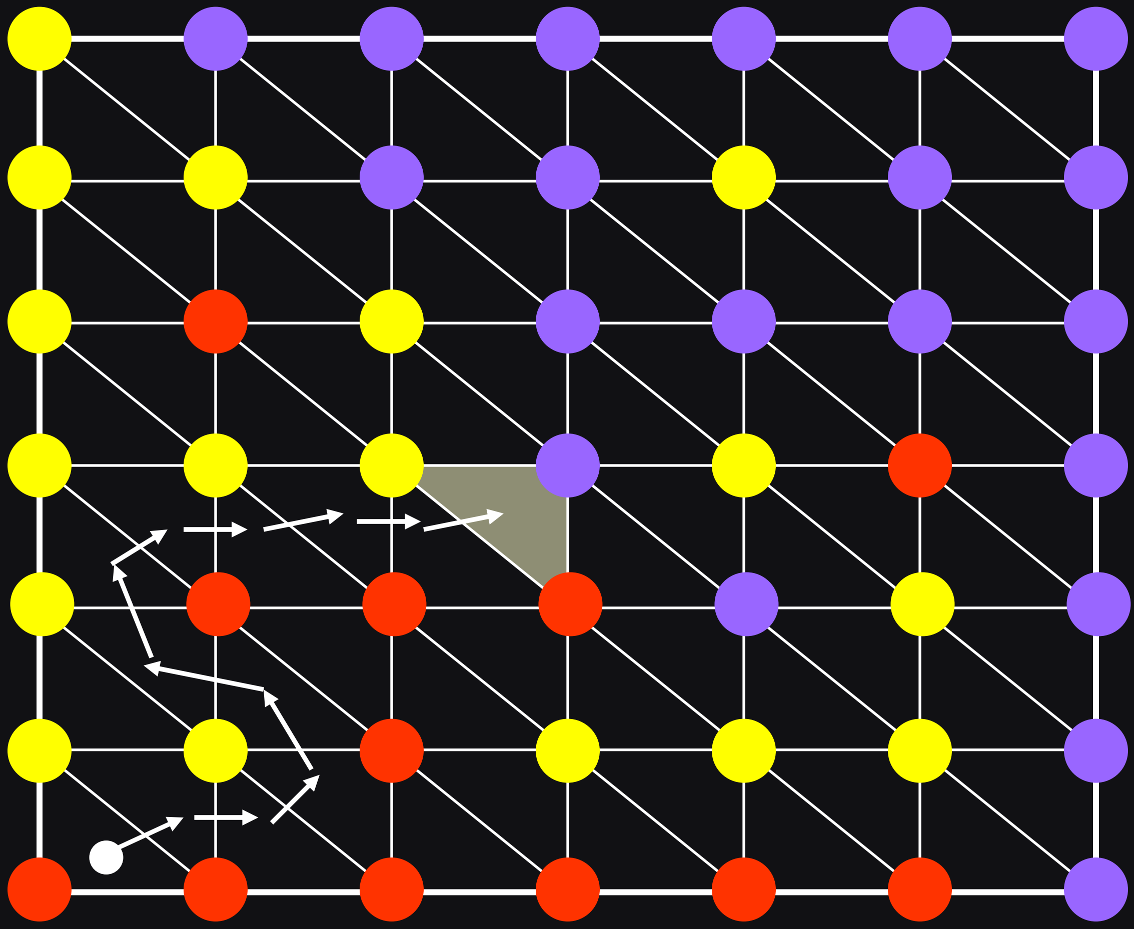
The SPERNER problem





SPERNER: Given C, find a trichromatic triangle.

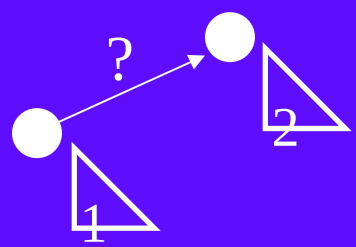
Solving SPERNER



(Abstract) Proof of Sperner's Lemma

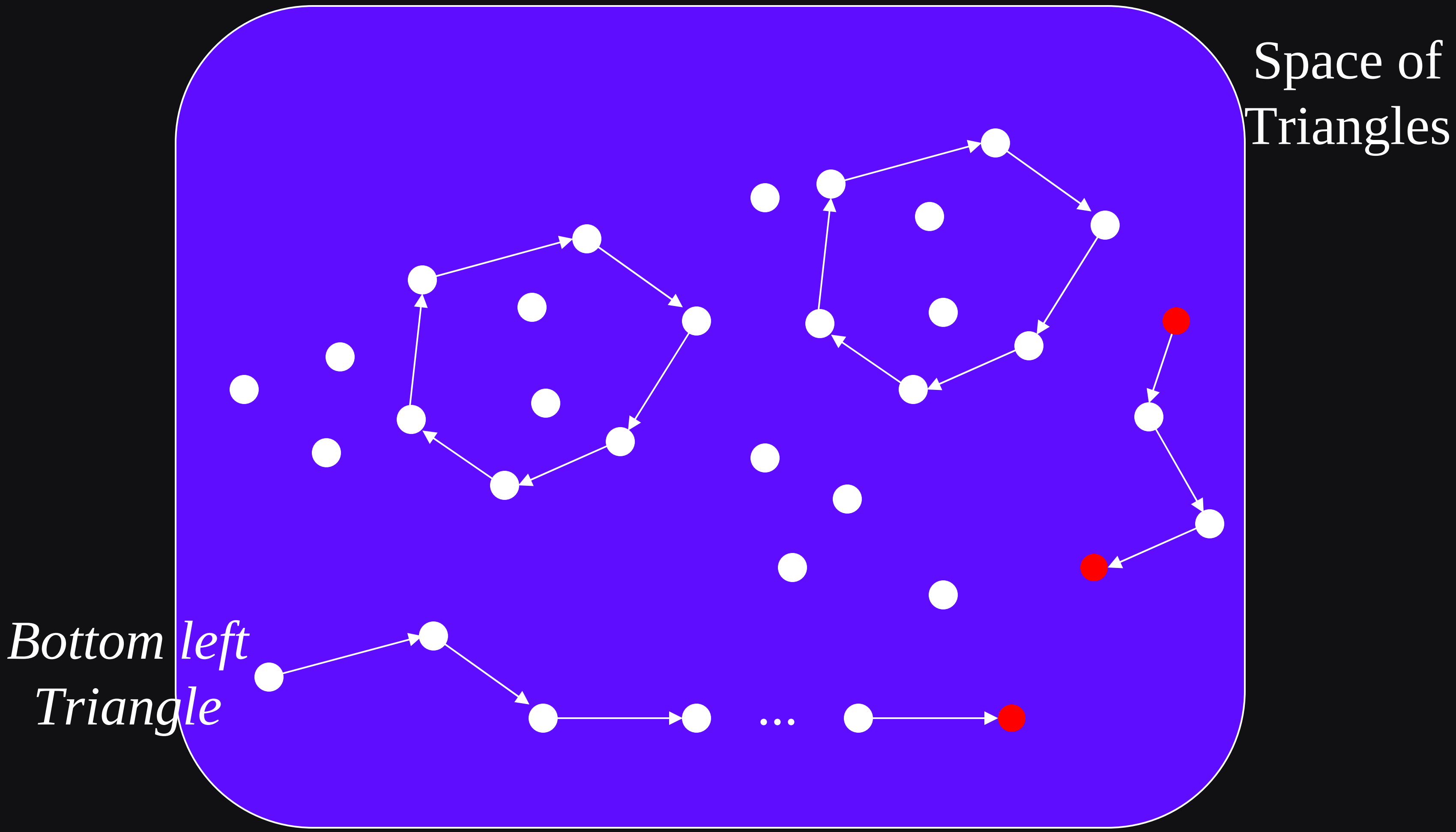
Space of Triangles

Transition Rule: *If* $\exists$ ***red*** - ***yellow*** *door cross it with* ***yellow*** *on your left hand*

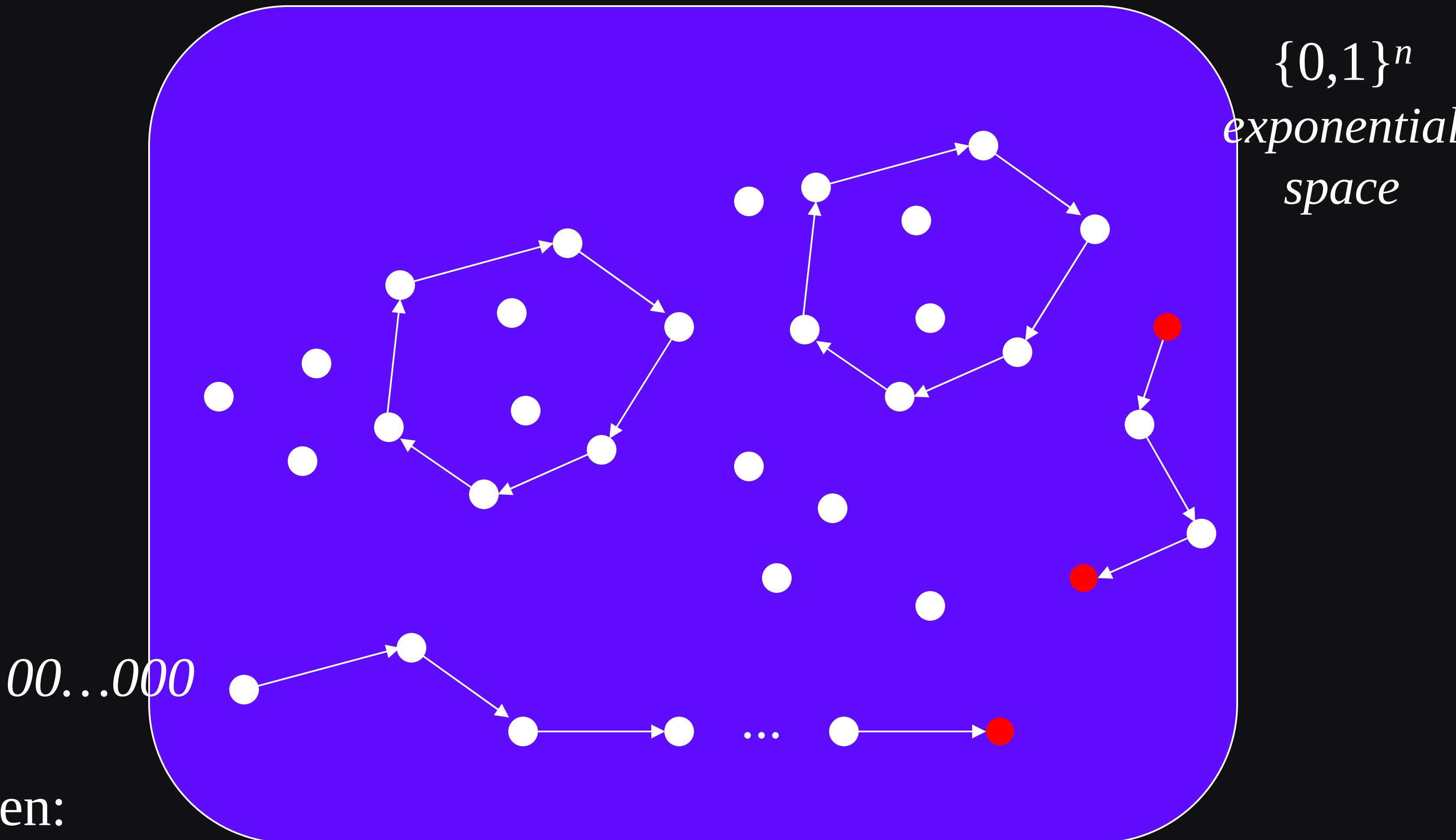


Lemma: No matter how the internal nodes are colored there exists a tri-chromatic triangle. In fact, an odd number of them.

(Abstract) Proof of Sperner's Lemma



(Abstract) SPERNER Problem



Given:

efficiently computable functions for finding ***next*** and ***previous***

Find:

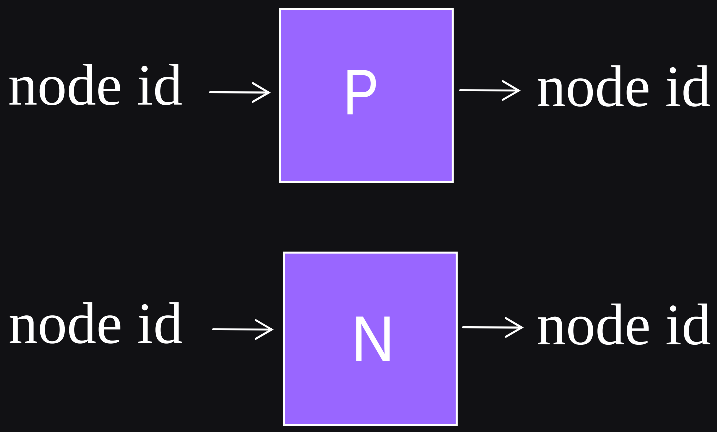
any terminal point different than 00…000

The PPAD Class [Papadimitriou ’94]

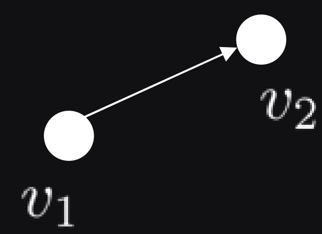
The class of all problems with guaranteed solution by dint of the following graph-theoretic lemma

> *A directed graph with an unbalanced node (node with indegree $\neq$ outdegree) must have another.*

Formally: a large graph is described by two circuits:

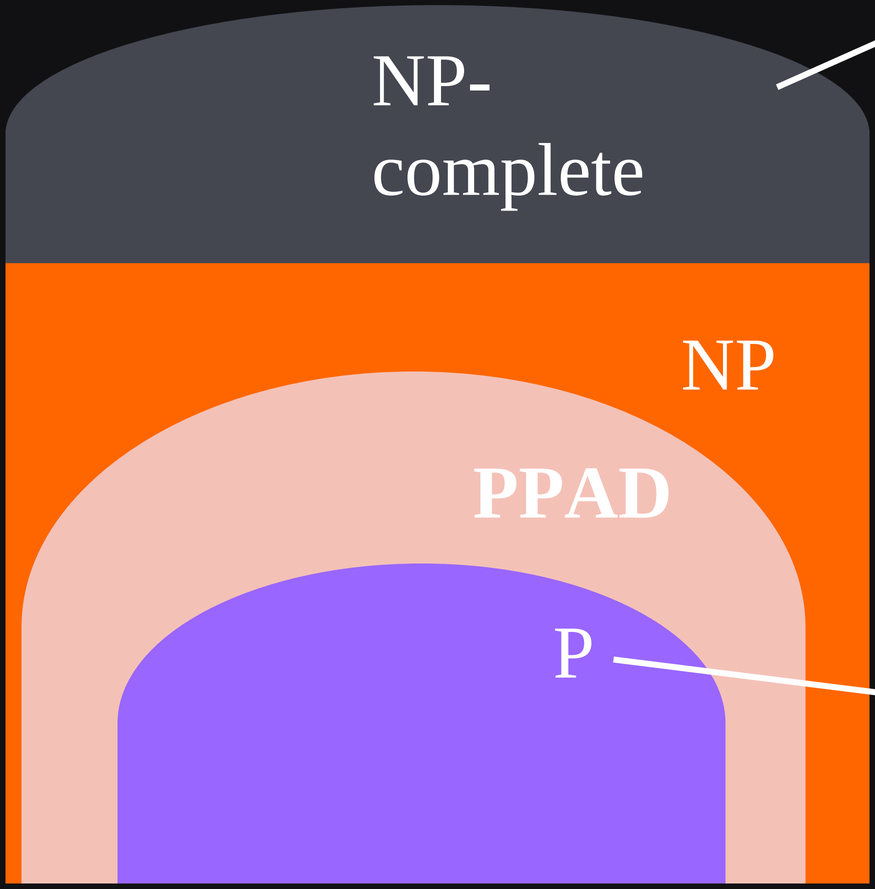


$$P(v_2) = v_1 \wedge N(v_1) = v_2$$



PPAD: *Given P and N, if 0^n is an unbalanced node, find another unbalanced node.*

Where is PPAD?



The hardest problems in NP
e.g.: quadratic programming
e.g.2: traveling salesman problem

Solutions can be found
in polynomial time
e.g.: linear programming
e.g.2: zero-sum games

Problems in PPAD

SPERNER $\in$ PPAD [Previous Slides]

BROUWER $\in$ PPAD [By Reduction to SPERNER-Scarf '67]

find an (approximately) fixed point of a continuous function from the unit cube to itself

SPERNER is PPAD-Complete [Papadimitriou '94]
[for 2D: Chen-Deng '05]

BROUWER is PPAD-Complete [Papadimitriou '94]

The Complexity of the Nash Equilibrium

Theorem:

Computing a Nash equilibrium is PPAD-complete…

- for games with ≥4 players;

  [Daskalakis, Goldberg, Papadimitriou ’05]

- for games with 3 players;

  [Chen, Deng ’05] & [Daskalakis, Papadimitriou ’0

- for games with 2 players.

  [Chen, Deng ’06]

Explaining the result

in 2-player games …

- there always exists a Nash eq. in rational numbers (why?)

- Lemke-Howson's algorithm 1964

➡ 2-NASH $\in$ PPAD

in $\geq$3-player games …

- there exists a 3-player game with only irrational Nash equilibria [Nash '51]

Computationally Meaningful NASH:

Given game $\mathcal{G}$ and ϵ, find an ϵ-Nash equilibrium of $\mathcal{G}$.

The Complexity of the Nash Equilibrium

Theorem:

Computing an ϵ-Nash equilibrium is PPAD-complete…

- for games with ≥4 players, $\epsilon \sim 2^{-n}$; *n*=#strategies;
  [Daskalakis, Goldberg, Papadimitriou '05]

- for games with 3 players, $\epsilon \sim 2^{-n}$; *n*=#strategies;
  [Chen, Deng '05] & [Daskalakis, Papadimitriou '0

- for games with 2 players, $\epsilon = 0$;
  [Chen, Deng '06]

Nash's Theorem "⇒" NASH ∈ PPAD

Nash $\longrightarrow$ Brouwer

Kick / Dive	Left	Right
Left	1 , -1	-1 , 1
Right	-1 , 1	1, -1

Penalty Shot Game

$\longmapsto$

f: $[0,1]^2 \rightarrow [0,1]^2$, cont.
such that
fixed point ≡ Nash eq.

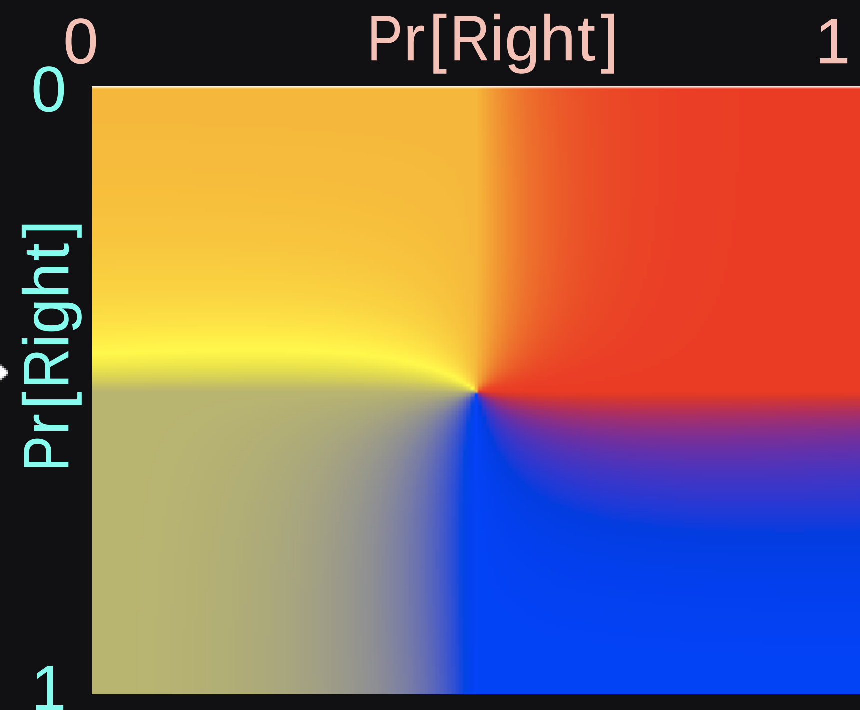
Nash's Theorem "$\Rightarrow$" NASH $\in$ PPAD

Nash $\longrightarrow$ Brouwer

Kick / Dive	Left	Right
Left	1 , -1	-1 , 1
Right	-1 , 1	1, -1

Penalty Shot Game

$\longmapsto$



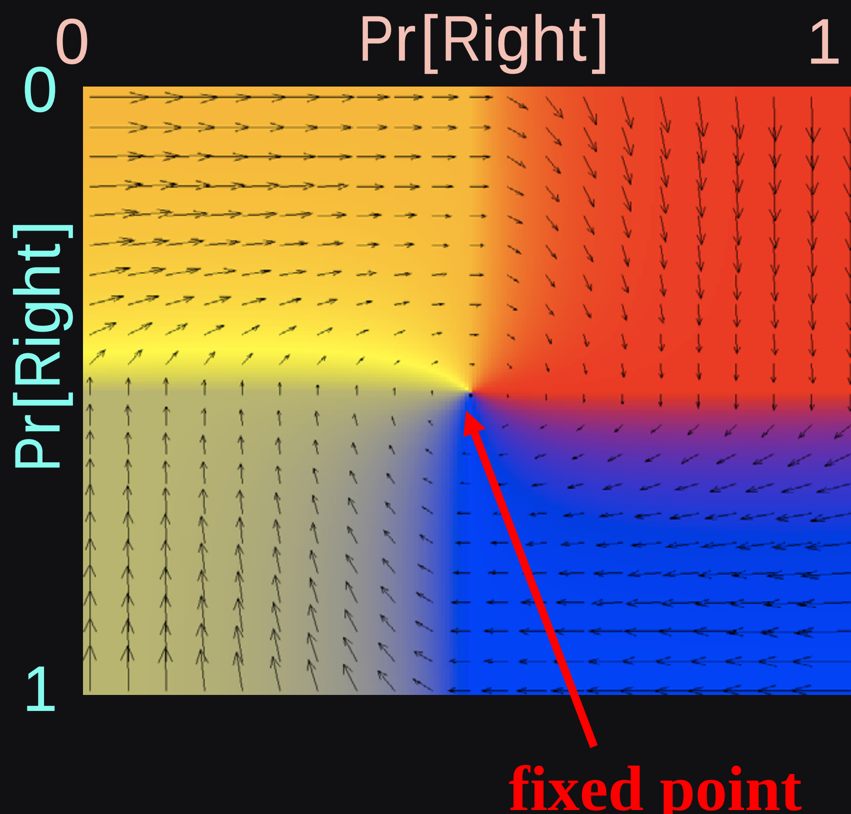
Nash's Theorem "⇒" NASH ∈ PPAD

Nash ⟶ Brouwer

		½	½
	Kick / Dive	Left	Right
½	Left	1 , -1	-1 , 1
½	Right	-1 , 1	1, -1

Penalty Shot Game

$\longmapsto$



Nash's Theorem "⇒" NASH ∈ PPAD

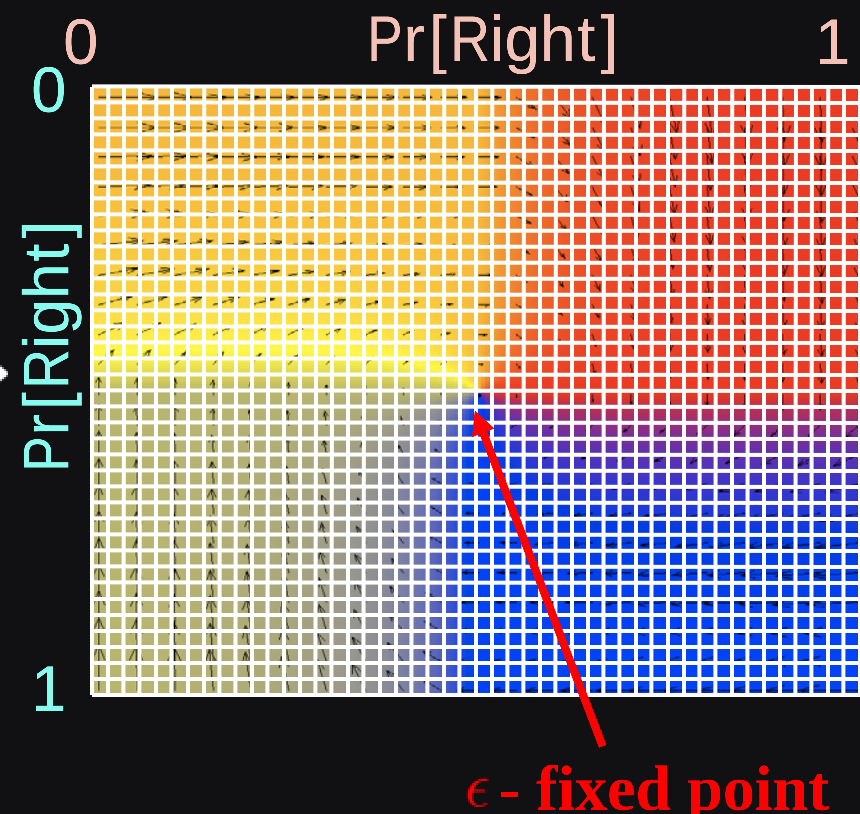
REAL PROOF

Nash $\longrightarrow$ Brouwer

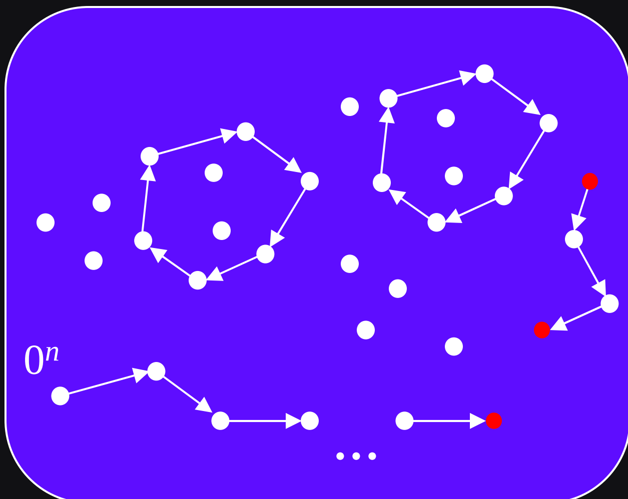
	½	½
Kick / Dive	Left	Right
½ Left	1 , -1	-1 , 1
½ Right	-1 , 1	1, -1

Penalty Shot Game

$\longmapsto$



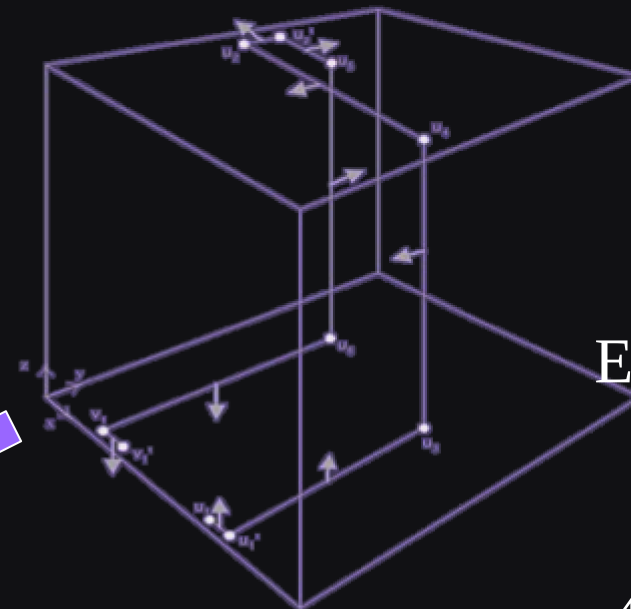
PPAD-hardness of NASH



Generic PPAD

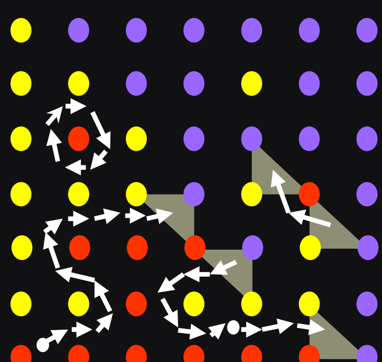
[Pap '94]

[DGP '05]



Embedded PPAD

[DGP '05]



SPERNER

[DGP '05]

p.w. linear BROUWER

[DGP '05]

multi-player NASH

[DGP '05]

4-player NASH

[DP '05]

[CD'05]

3-player NASH

[CD'05]

2-player NASH

PPAD-Hardness of NASH [DGP '05]

Nash $\longleftarrow$ Brouwer

game whose Nash equilibria are close to the fixed points of f $\longleftarrow\!\shortmid$ f: $[0,1]^3 \rightarrow [0,1]^3$, continuous & p.w.linear

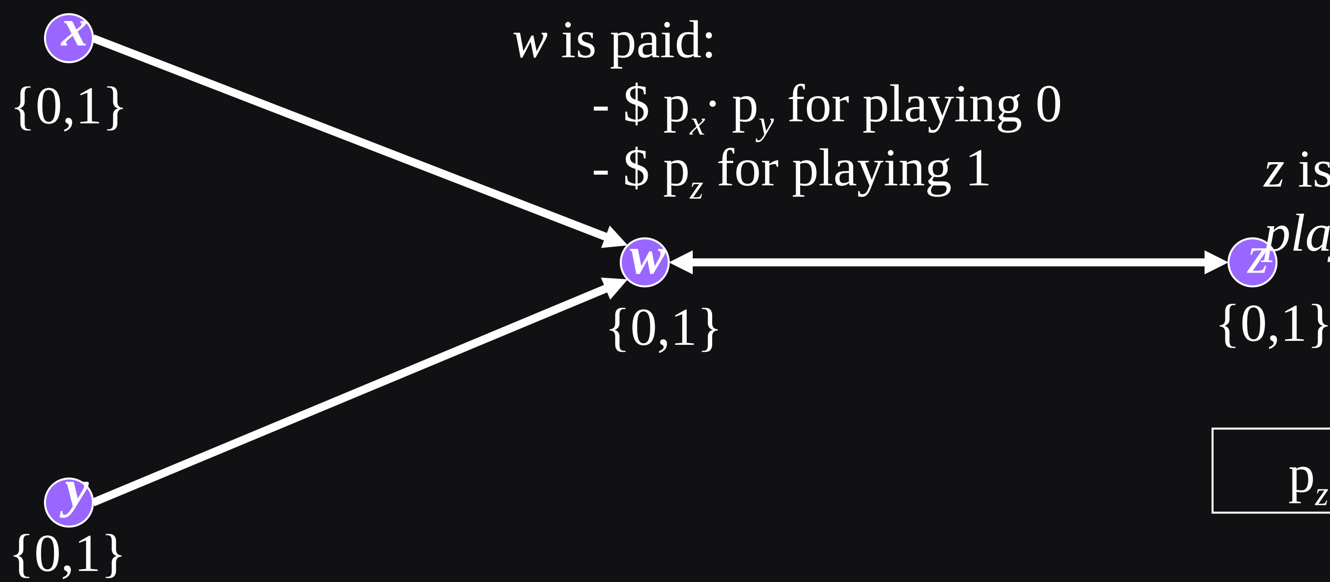
- *Game-gadgets*: games acting as arithmetic gates

Games that do real arithmetic

e.g. multiplication game *(similarly addition, subtraction)*

two strategies per player, say {0,1};

➡ Mixed strategy ≡ a number in [0,1]
(probability of playing 1)



w is paid:
- \$ $p_x \cdot p_y$ for playing 0
- \$ p_z for playing 1

z is paid $1-p_w$ *for playing 1*

$\boxed{p_z = p_x \cdot p_y}$

Games that do real arithmetic

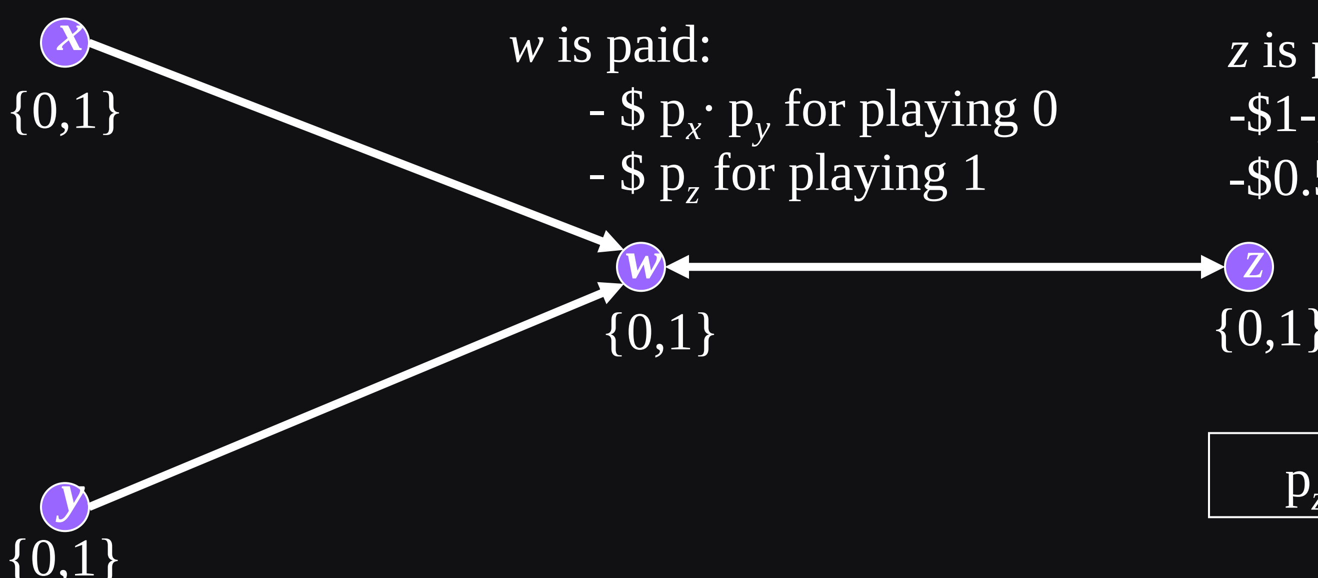
w's payoff

for playing 0

	y plays 0	y plays 1
x plays 0	0	0
x plays 1	0	1

for playing 1

z plays 0	0
z plays 1	1



w is paid:

- \$ $p_x \cdot p_y$ for playing 0
- \$ p_z for playing 1

z is paid:

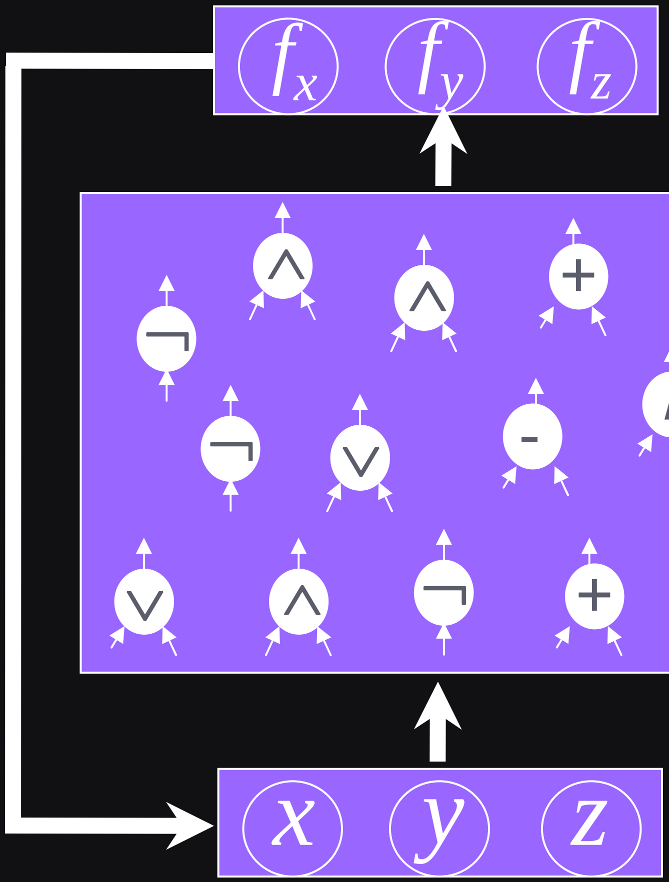
-\$1-p_w for playing 1

-\$0.5 for playing 0

$$\boxed{p_z = p_x \cdot p_y}$$

PPAD-Hardness of NASH [DGP '05]

Nash $\longleftarrow$ Brouwer

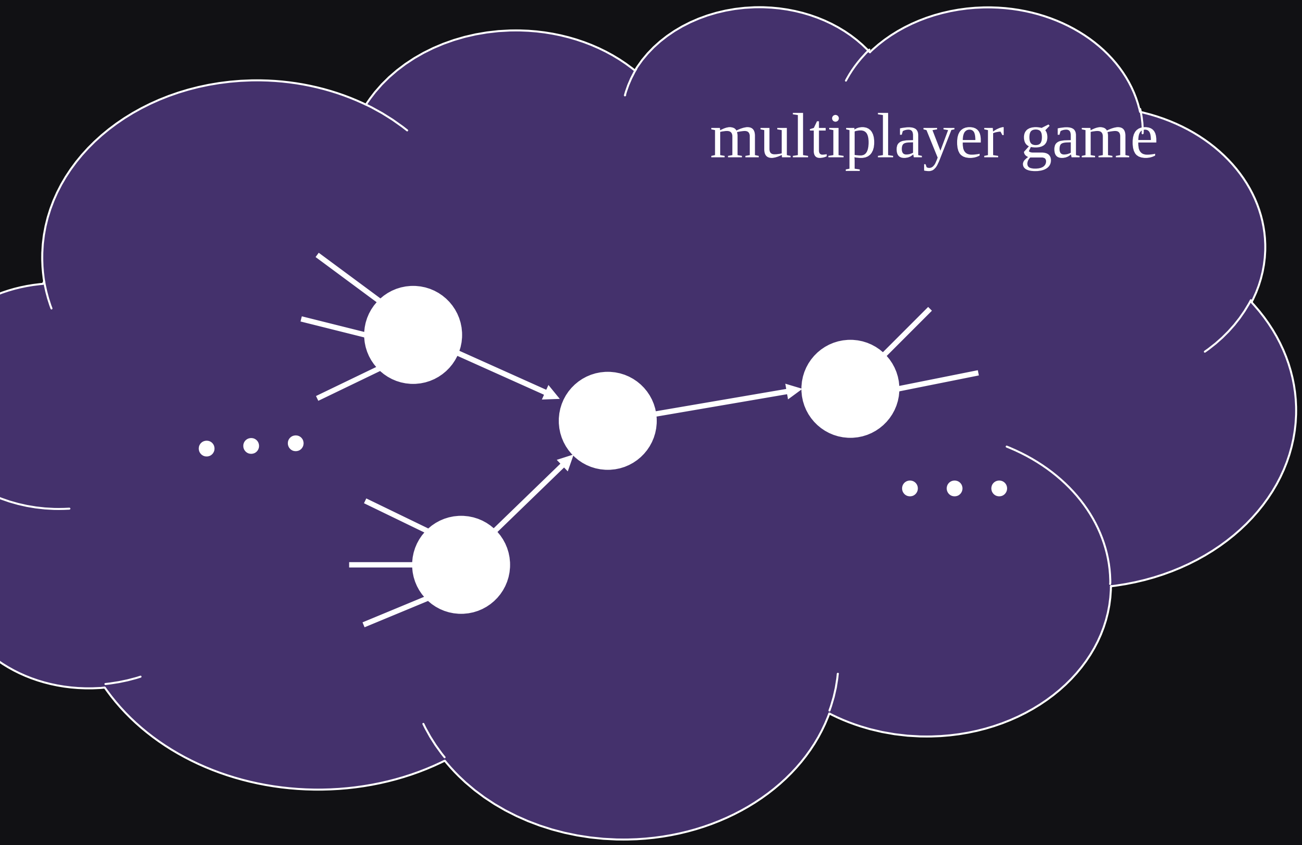


$\longleftarrow\!\!\dashv$

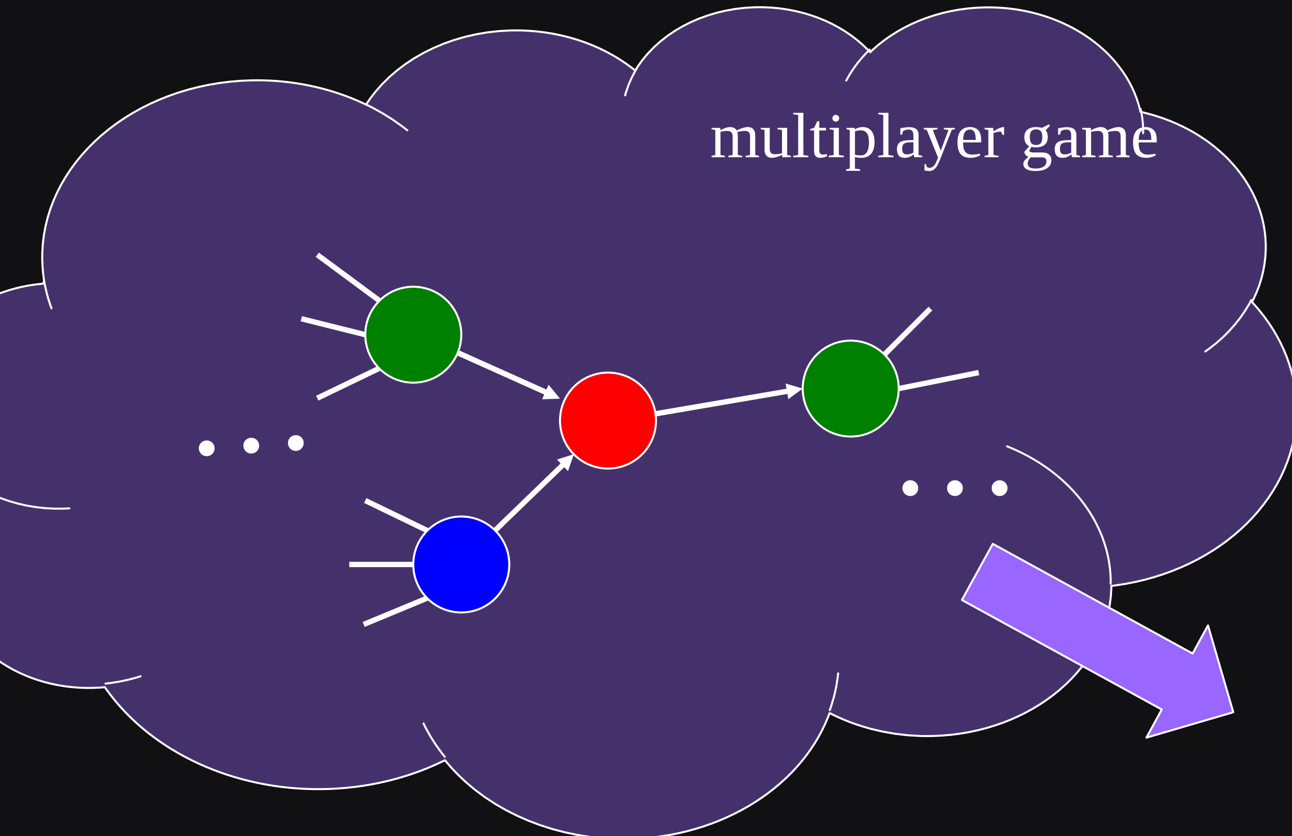
f: $[0,1]^3 \rightarrow [0,1]^3$,
continuous & p.w.linear

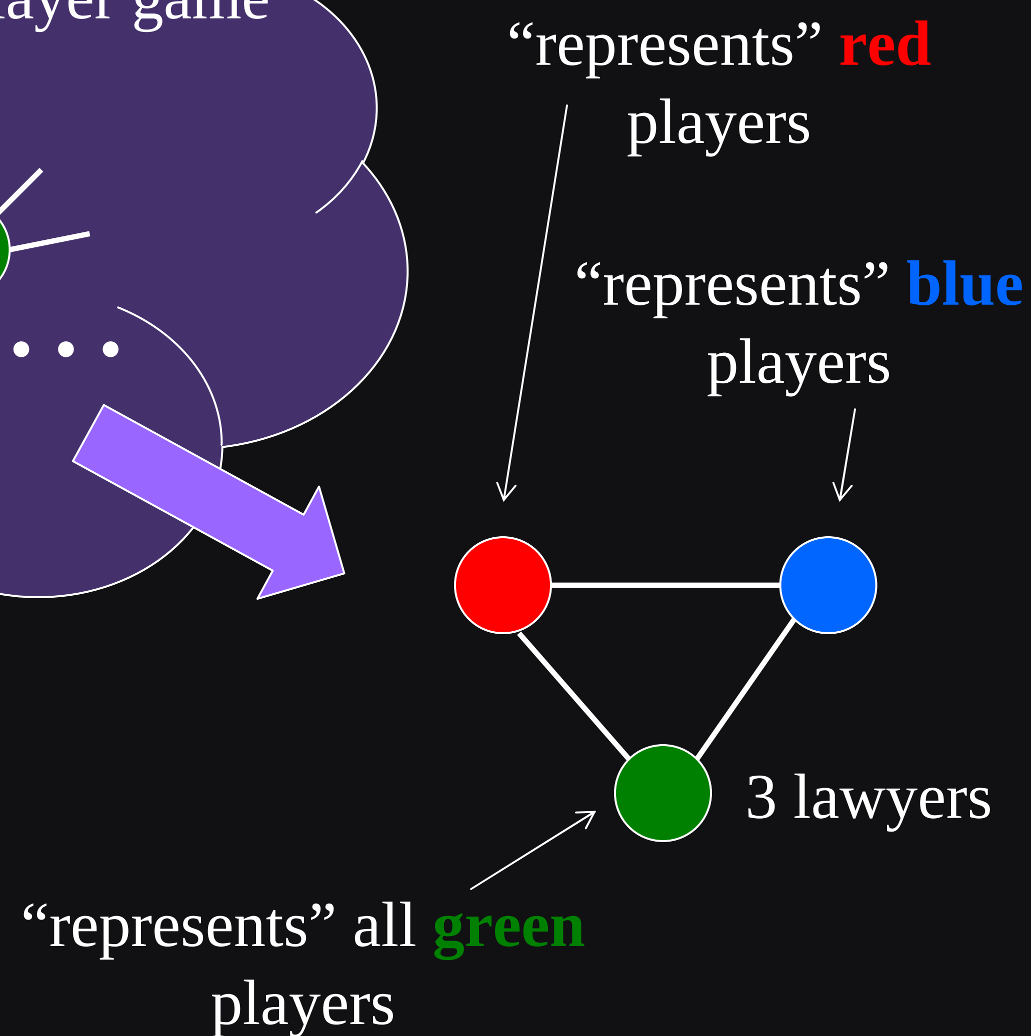
- use *game-gadgets* to simulate f with a game
- *Topology*: noise reduction

Reduction to 3 players [Das, Pap '05]



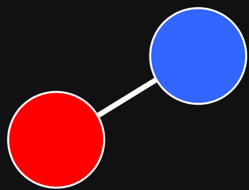
Reduction to 3 players [Das, Pap '05]



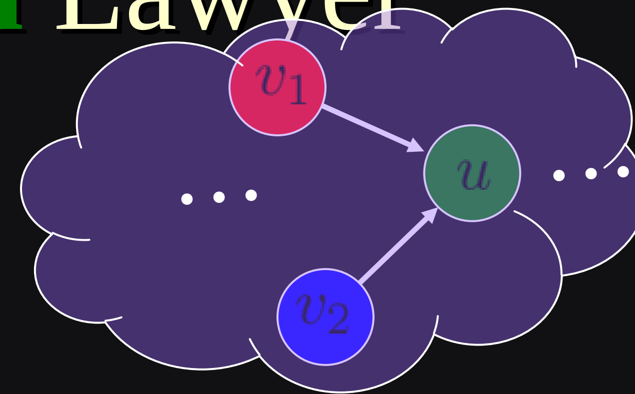


Coloring: no two nodes affecting one another, or affecting the same third player use the same color;

Payoffs of the Green Lawyer



	$v_2 : 0$	$v_2 : 1$	$\neq v_2$
$v_1 : 0$	copy of the payoff table of node *u*		0
$v_1 : 1$			0
$\neq v_1$	0	0	0

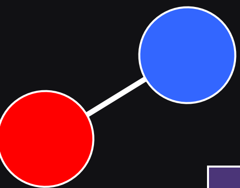


payoffs of the **green** lawyer for representing node *u*

wishful thinking: The Nash equilibrium of the lawyer-game, gives a Nash equilibrium of the original multiplayer game, after marginalizing with respect to individual nodes.

But why would a lawyer represent every node equally?

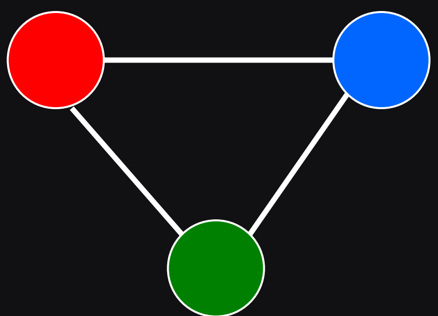
Enforcing Fairness



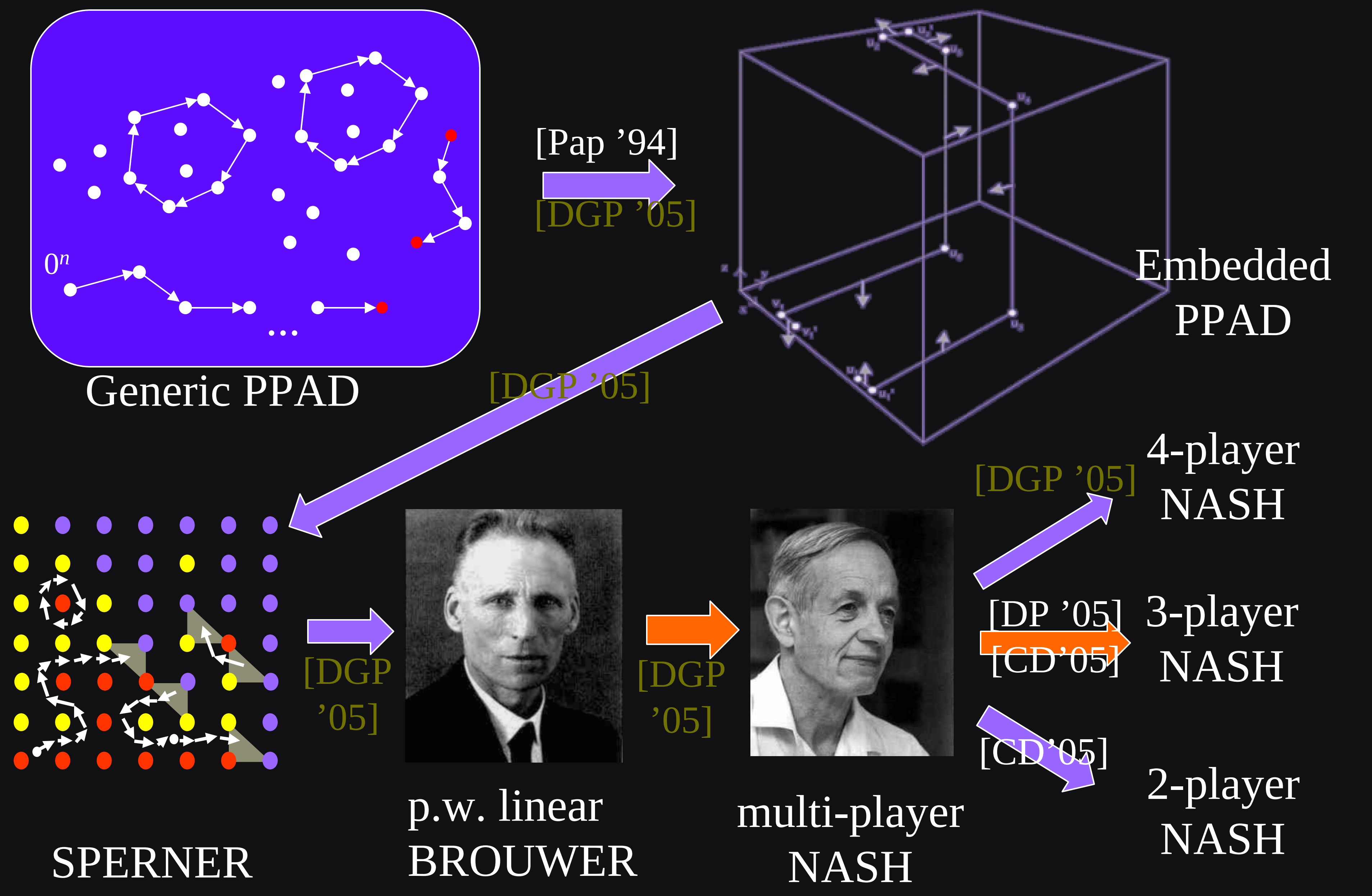
	$v_2 : 0$	$v_2 : 1$	$\neq v_2$
$v_1 : 0$	copy of the payoff table of node *u*		0
$v_1 : 1$			0
$\neq v_1$	0	0	0

$+$

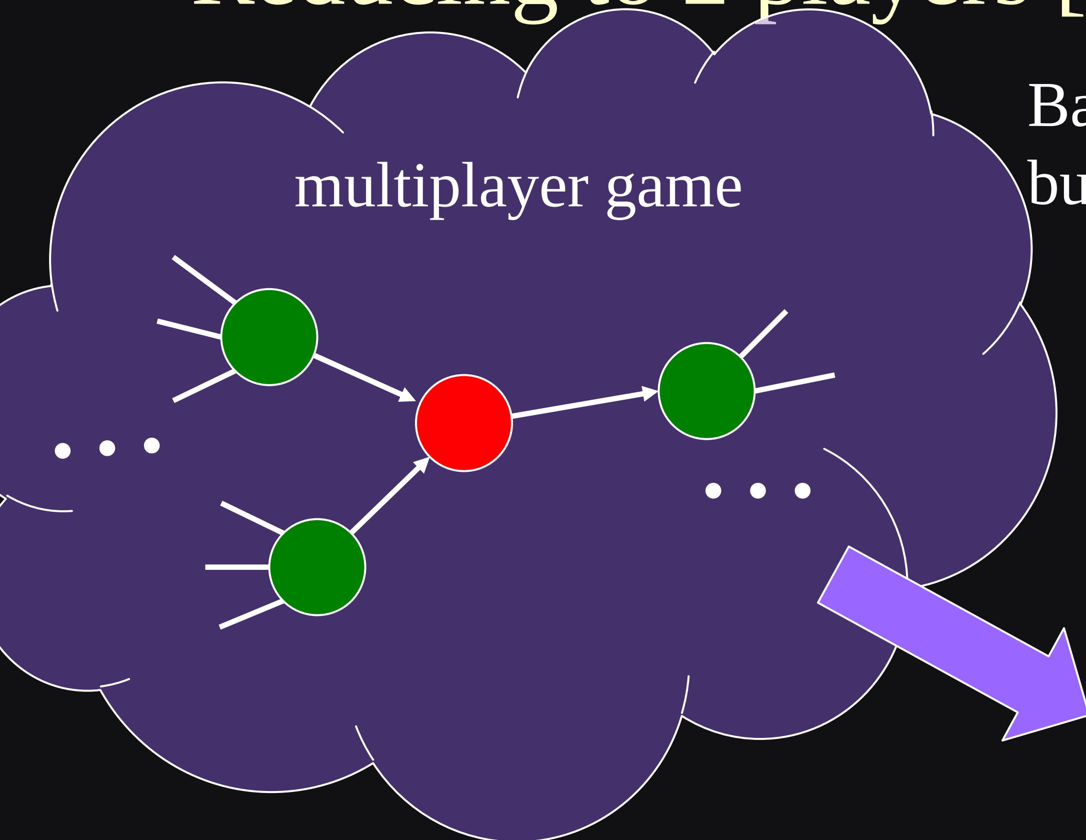
lawyers play on the side a high-stakes game over the nodes they represent



PPAD-hardness of NASH



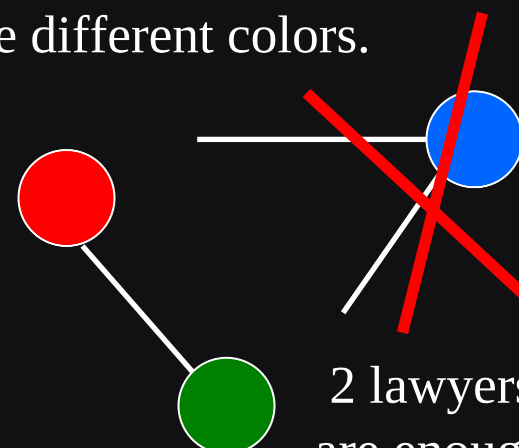
Reducing to 2 players [Chen, Deng ’05]



Based on the following simple, but crucial observation:

- the expected payoff of each lawyer is additive w.r.t. the nodes that another lawyer represents;

- hence, if two nodes affect the same third node, they don’t need to have different colors.



2 lawyers are enough

Coloring: no two nodes affecting one another, ~~or affecting the same third player~~ use the same color;

two colors suffice to color the multiplayer game in the [DGP 05] construction

Recapping

[Nash '51]: NASH $\leq_p$ BROUWER.

[D. Gold. Pap. '05]: BROUWER $\leq_p$ NASH. (i.e. NASH is PPAD-complete)

[Chen, Deng '06]: ditto for 2-player games.

NASH: Given game $\mathcal{G}$ and error ϵ, find an ϵ-Nash equilibrium of $\mathcal{G}$.

Above results hold for $\epsilon \leq 2^{-\Omega(n)}$, where n is the #strategies.

[Chen, Deng, Teng '06] : $(n^{-\alpha})$ - NASH is also PPAD-*complete*.

Constant ε 's?

[Lipton, Markakis, Mehta '03]:

*For any ϵ, an **additive** ϵ- Nash equilibrium can be found in time $n^{O(\log n/\epsilon^2)}$.*

(Hence, it is unlikely that additive ϵ-NASH is PPAD-complete, for constant values of ϵ.*)*

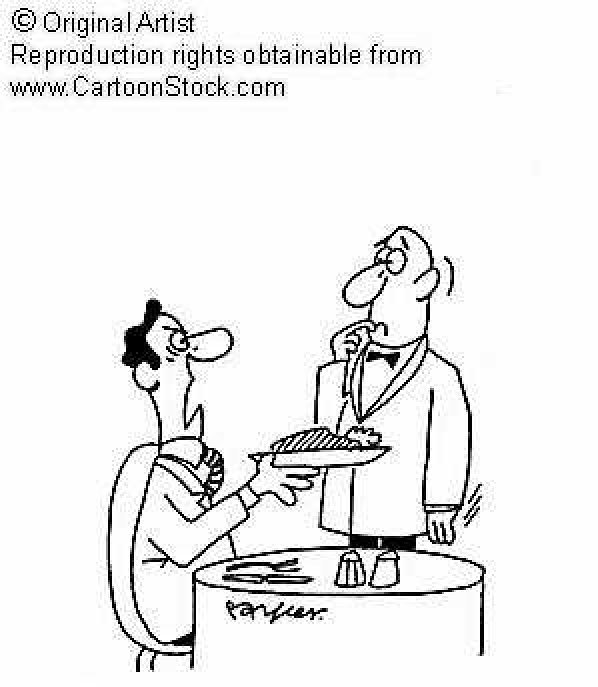
Efficient Algorithms: ϵ = .75 → .50 → .38 → .37 → .34 [Tsaknakis, Spirakis '08]

The trouble with approximate Nash

Algorithms expert to TSP user:

Unfortunately, with current technology we can only give you a solution guaranteed to be no more than 50% above the optimum

The trouble with approximate Nash (cont.)



Irate Nash user to algorithms expert:

Why should I adopt your recommendation and refrain from acting in a way that I know is much better for me? And besides, given that I have serious doubts myself, why should I even believe that my opponent(s) will adopt your recommendation?

Bottom line

- PTAS is the only interesting question here…

And what about relative approximations?

Recall, relative approximation: Payoff $\geq (1 - \varepsilon)$ OPT;

Result of Lipton-Markakis-Mehta does not hold anymore;

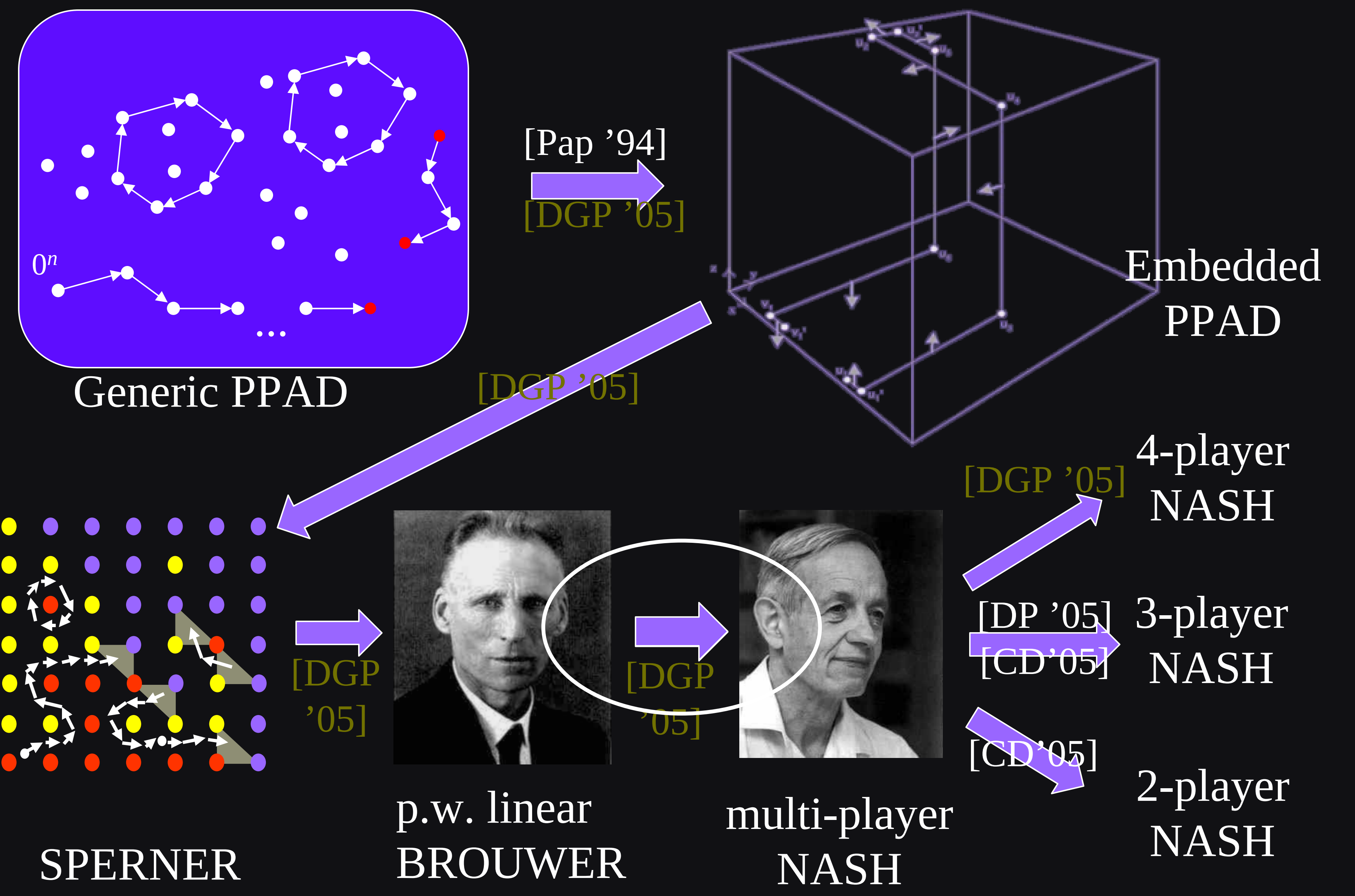
Hot of the press [Daskalakis '09]:

Relative ε-NASH *is* PPAD-*complete, even for constant* ε*'s.*

Challenges: 1. gadgets in [DGP '05] do not work for constant ε*'s*; we redo the construction introducing some kind of "gap amplification" gadget;

2. the high-stakes lawyer-game overwhelms the payoffs of the multiplayer game if we look at relative approximations with constant ε*'s*…

Future PPAD-hardness reductions



part 2a

Equilibrium Computation in Compactly-Represented Games

Costis Daskalakis & Kevin Leyton-Brown

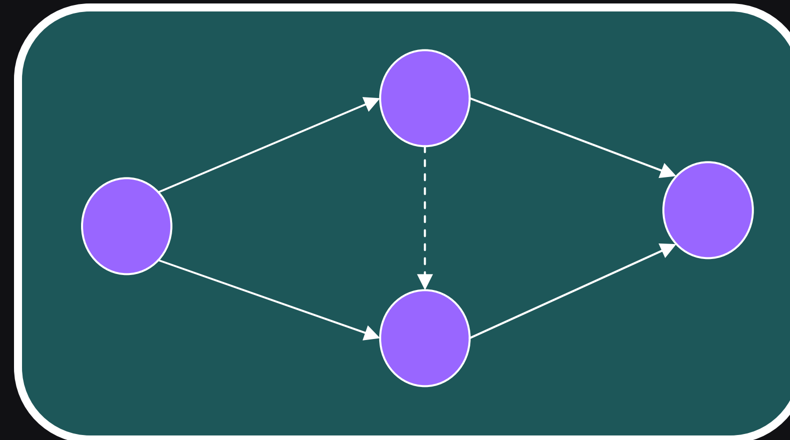
Part 2(a)

“If your game is interesting, then its description cannot be astronomically long.”

Christos Papadimitriou

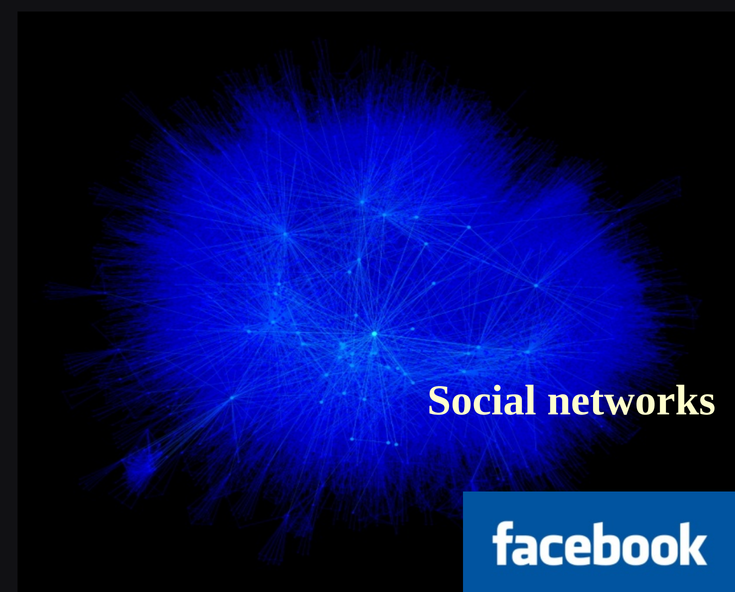
Markets

Internet routing



Evolution

Elections



Social networks

Computationally motivated compact representations

- normal form game description can be very wasteful;
  (if n players, s strategies, description size is $n\,s^n$)

- it is possible that by further exploiting the structure of the game, the game can be described more efficiently;

- in this part of the tutorial, we investigate succinct game-representations which allow certain large games to be compactly described and also make it possible to efficiently find an equilibrium

Games of Polynomial Type

- The normal form representation lists explicitly everybody's name, action space, and payoffs;

-A first step towards a generalization:

A game description is called *of polynomial type*, if

- the number of players is polynomial in the description size;
- the number of actions available to each player is polynomial in the description size;
- BUT no requirement to list every payoff explicitly;

e.g. 1: (polynomial type) normal-form games, rest of this session…

e.g. 2: (non polynomial type) poker, traffic.

The Expected Utility Problem

- A game description specifies the payoff of a player, given the other players' actions.

- How hard is it to compute a player's expected payoff given the mixed strategies of the other players?

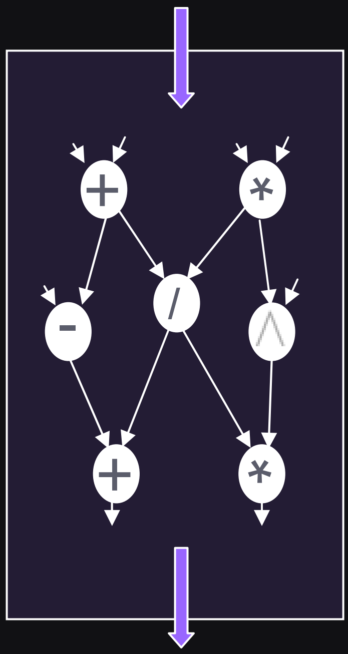
e.g. 1 (easy case) Normal form games

e.g. 2 (hard case) Suppose every player has two strategies 0/1, and given everybody's strategy a circuit C_i ,

$$C_i : \{0,1\}^n \to \mathbb{R},$$

computes player *i*'s payoff.

Compactness pays off



Theorem [Daskalakis, Fabrikant, Papadimitriou ’06]

If a game representation is of polynomial type and the expected utility problem can be solved by a polynomially long arithmetic circuit using +,-,*,/, max, min (i.e. a straight-line program), then finding a mixed Nash equilibrium is in PPAD.

Remark: Can be generalized to non polynomial-type games such as extensive-form games, congestion games; see [DFP ’06].

Theorem [Papadimitriou ’05]

If a game representation is of polynomial type and the expected utility problem can be solved by a polynomial-time algorithm, then finding a correlated equilibrium is in P.

Symmetries in Games

Symmetric Games: Each player p has

- the same set of strategies

$$S = \{1, \ldots, s\}$$

$\boxed{\textit{Size: } s\, n^{s-1}}$

- the same payoff function $u = u(\underbrace{\sigma}_{\text{choice of } p} ; \underbrace{n_1, n_2, \ldots, n_s}_{\text{number of the other players choosing each strategy in } S})$

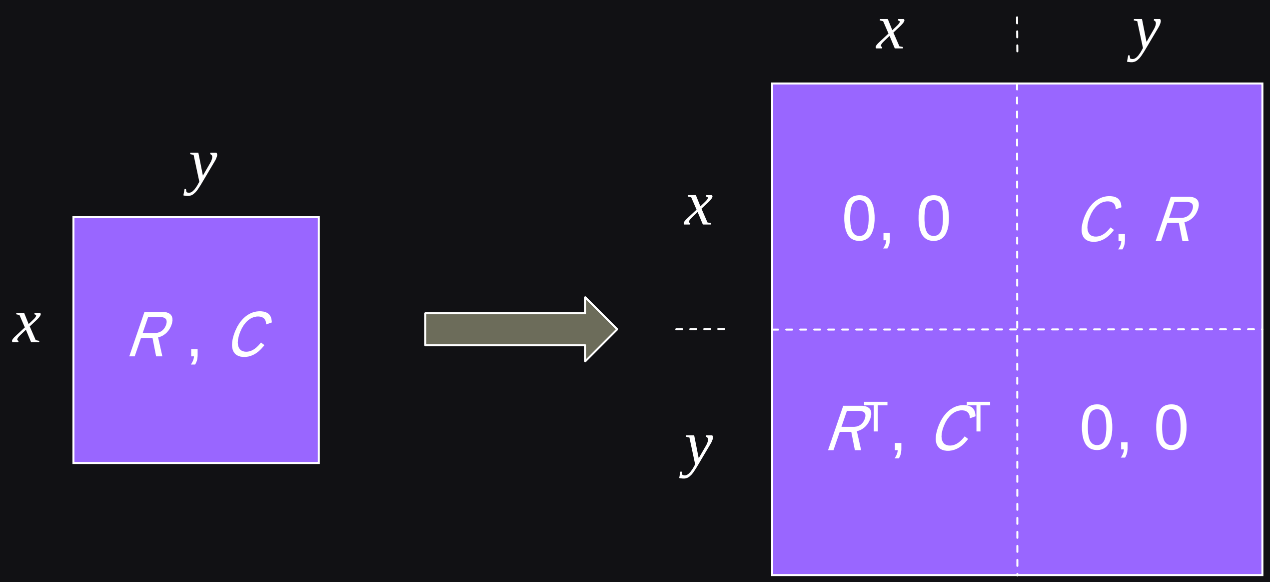
***E.g.* :** - Rock-Paper-Scissors

- traffic (congestion) games, with same source destination pairs for each player

Nash '51: Always exists an equilibrium in which every player uses the same mixed strategy

Symmetrization

[Gale-Kuhn-Tucker 1950]

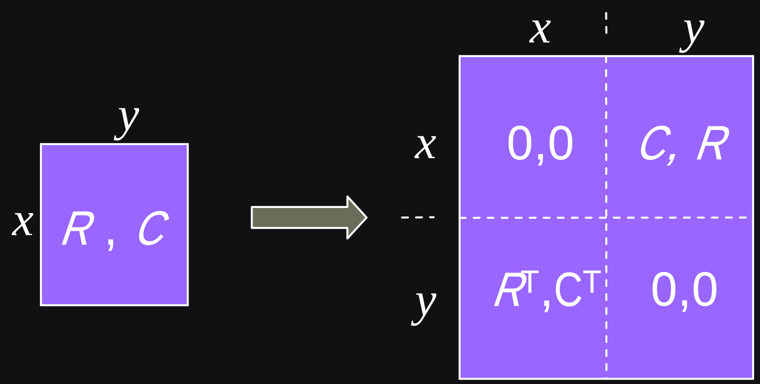


Equilibrium ⟸ Symmetric Equilibrium

In fact […]

Equilibrium ⟸ Any Equilibrium

Symmetrization



Hence, PPAD to solve symmetric 2-player games

Equilibrium $\Leftarrow$ Symmetric Equilibrium

In fact […]

Equilibrium $\Leftarrow$ Any Equilibrium

Open: - Reduction from 3-player games to symmetric 3-player games

- Complexity of symmetric 3-player games

Multi-player symmetric games

If n is large, s is small, a symmetric equilibrium

$$\mathrm{x} = (x_1, x_2, \ldots, x_s)$$

can be found as follows:

- guess the support of x, 2^s possibilities

- write down a set of polynomial equations an inequalities corresponding to the equilibrium conditions, for the guessed support

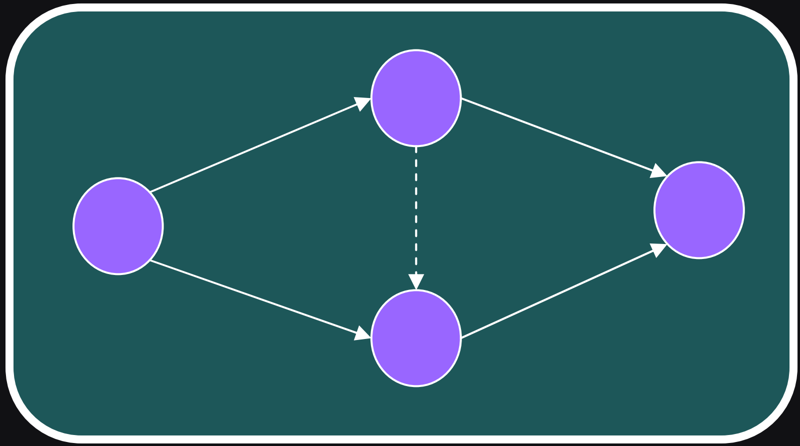
- polynomial equations and inequalities of degree n in s variables

can be solved approximately in time
$n^s \log(1/\varepsilon)$

how far with symmetric games?

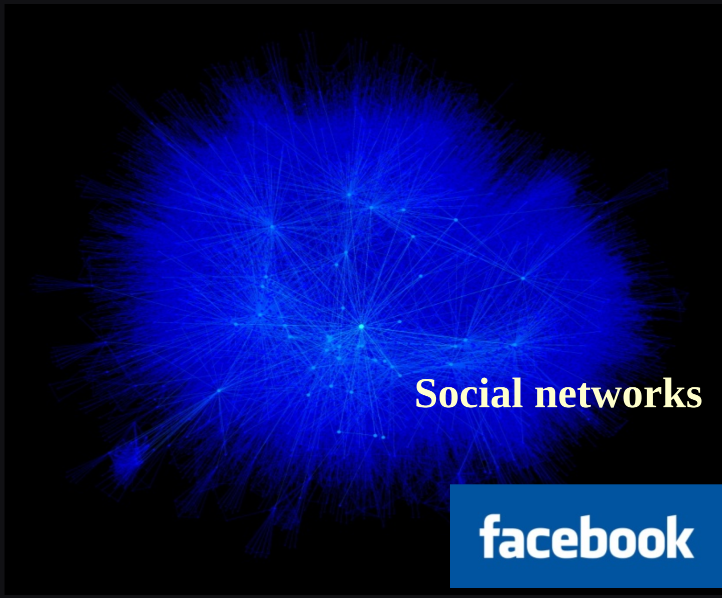
Markets

Internet routing



Evolution

Elections



anonymous games

Every player is (potentially) different, but only cares about how many players (of each type) play each of the available strategies.

e.g. symmetry in auctions, congestion games, social phenomena, etc.

‘‘Congestion Games with Player- Specific Payoff Functions.’’
Milchtaich, *Games and Economic Behavior, 1996.*

‘‘The women of Cairo: Equilibria in Large Anonymous Games.’’
Blonski, *Games and Economic Behavior*, 1999.

“Partially-Specified Large Games.”
Ehud Kalai, *WINE*, 2005.

reasons for anonymous

- *ubiquity*: much richer than symmetric games

  think of your favorite large game - is it anonymous?

- *succinctness*: not nearly as wasteful as general normal form games

  n players, s strategies, all interact, n^s description (rather than ns^n)

  (the utility of a player depends on her strategy, and on how many other players play each of the s strategies)

- *robustness*:

  Nash equilibira of the simultaneous move game are robust with regards to the details of the game (order of moves, information transmission, opportunities to revise actions etc. [Kalai '05])

working assumption: n large, s small (o.w. PPAD-Complete)

PTAS for anonymous

Theorem: *If the number of strategies* s *is a constant,*
[with Pap. '07, '08] *there is a PTAS for mixed Nash equilibria.*

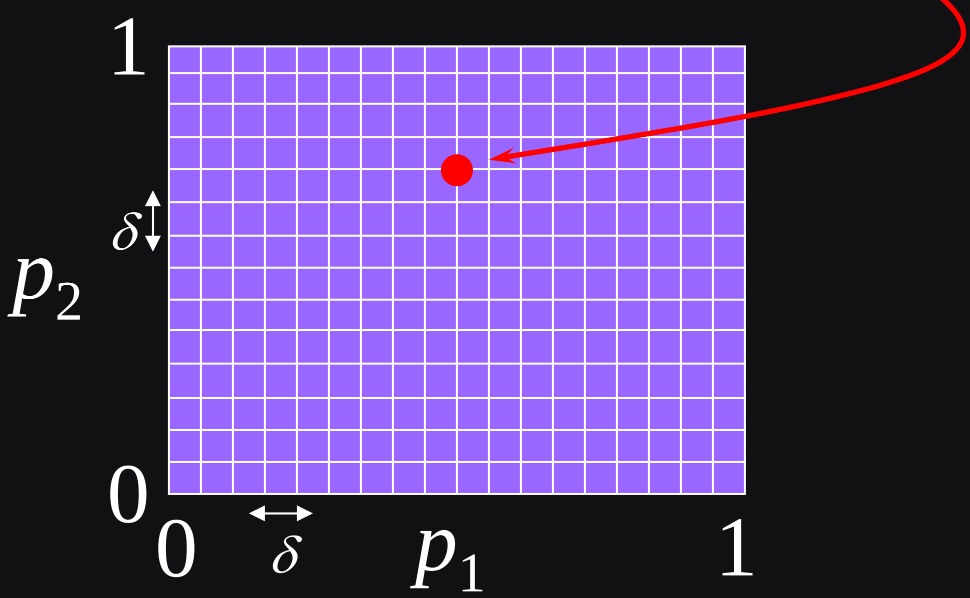
Remarks: - exact computation is not known to be PPAD-complete

- if *n* is small and *s* is large (few players many strategies) then PPAD-complete

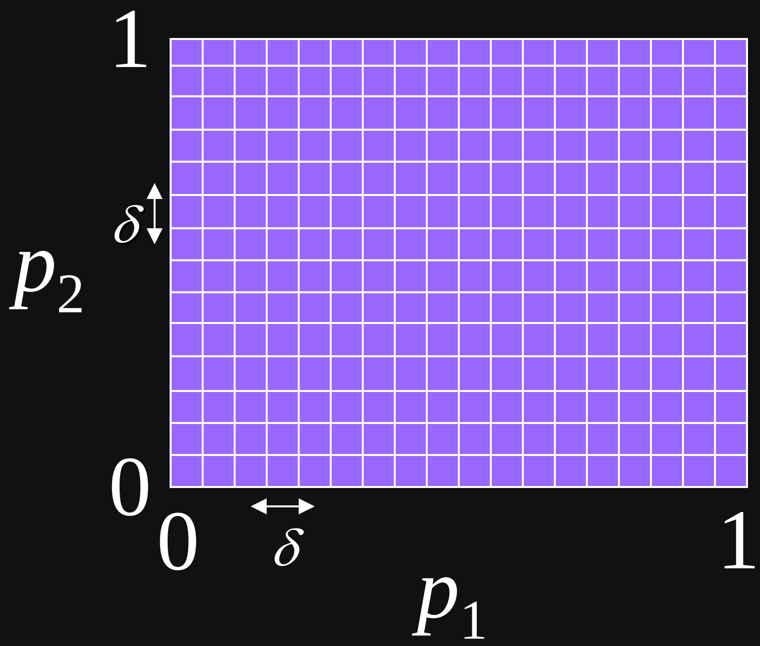
sketch for 2 strategies

Masterplan:

- since 2 strategies per player, Nash eq. lies in $[0,1]^n$
- discretize $[0,1]^n$ into multiples of δ, and restrict search to the discrete space
- pick best point in discrete space



sketch for 2 strategies (cont.)



*F*irst trouble:

size of search space $\left(\frac{1}{\delta}\right)^n$

by exploiting anonymity
(max-flow argument) $\left(n\right)^{1/\delta}$

Basic Question:

what grid size δ is required for ε - approximation?

if function of ε only $\Rightarrow$ PTAS

if function also of $n \Rightarrow$ nothing

sketch for 2 strategies (cont.)

Theorem [D., Papadimitriou '07]:

Given

- n ind. Bernoulli's X_i with expectations p_i, i =1,…, n
- a constant δ independent of n

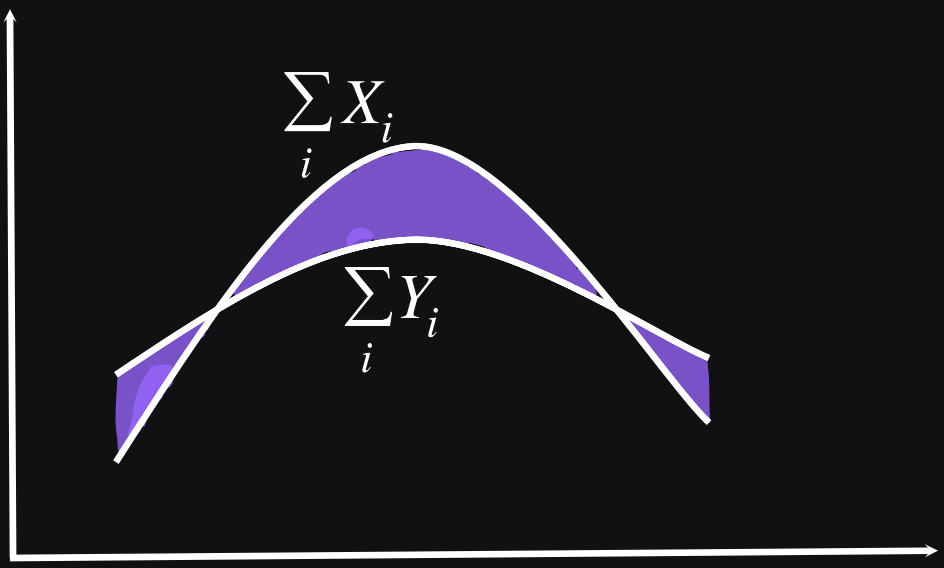
there exists another set of Bernoulli's Y_i with expectations q_i such that

q_i's are integer multiples of δ

$$\left\| \sum_i X_i - \sum_i Y_i \right\|_{\mathrm{TV}} \leq O(\sqrt{\delta})$$

total variation distance cheat sheet

$$\left\| \sum_i X_i - \sum_i Y_i \right\|_{\mathrm{TV}}$$



$$\sum_{t=0}^{n} \left| \Pr\left[\sum_i X_i = t\right] - \Pr\left[\sum_i Y_i = t\right] \right| = \ell_1\left(\sum_i X_i, \sum_i Y_i\right)$$

sketch for 2 strategies (cont.)

the Nash equilibrium

Theorem [D., Papadimitriou ’07]:

the grid size

Given

- n ind. Bernoulli’s X_i with expectations p_i , i =1,…, n
- a constant δ independent of n

there exists another set of Bernoulli’s Y_i with expectations q_i such that

q_i’s are integer multiples of δ

$$\left\| \sum_i X_i - \sum_i Y_i \right\|_{\mathrm{TV}} \leq O(\sqrt{\delta}) \quad \Rightarrow$$

ε - approximation in time $n^{O(1/\varepsilon^2)}$

regret if we replace the X_i’s by the Y_i’s

proof of approximation result

- rounding p_i's to the closest multiple of δ *gives total variation* $n\delta$
- probabilistic rounding up or down quickly runs into problems
- what works:

Law of Rare Events ← *Poisson Approximations (Stein's Method)*

$+$

CLT ← *Berry-Esséen*

proof of approximation result

Intuition:

If p_i's were small $\Rightarrow$ $\sum_i X_i$ would be close to a Poisson with mean $\sum_i p_i$

$\Rightarrow$ define the q_i's so that $\sum_i q_i \approx \sum_i p_i$

$$\begin{tikzcd}
\sum_i X_i \arrow[d, leftrightarrow] & & \sum_i Y_i \arrow[d, leftrightarrow] \\
\mathit{Poisson}\left(\sum_i p_i\right) \arrow[rr, leftrightarrow] & & \mathit{Poisson}\left(\sum_i q_i\right)
\end{tikzcd}$$

proof of approximation result

Poisson approximation is only good for small values of p_i's. (LRE)

For intermediate values of p_i's, Normals are better. (CLT)

$$
\begin{tikzcd}
\sum_i X_i \arrow[d, leftrightarrow, "\textit{Berry-Esséen}"] & & \sum_i Y_i \arrow[d, leftrightarrow, "\textit{Berry-Esséen}"] \\
\mathcal{N}\left(\sum_i p_i, \sum_i p_i(1-p_i)\right) \arrow[rr, leftrightarrow] & & \mathcal{N}\left(\sum_i q_i, \sum_i q_i(1-q_i)\right)
\end{tikzcd}
$$

binomial approximation result

the Nash equilibrium

Theorem [D., Papadimitriou '07]:

Given

the grid size

- n ind. Bernoulli's X_i with expectations p_i , i =1,…, n
- a constant δ independent of n

there exists another set of Bernoulli's Y_i with expectations q_i such that

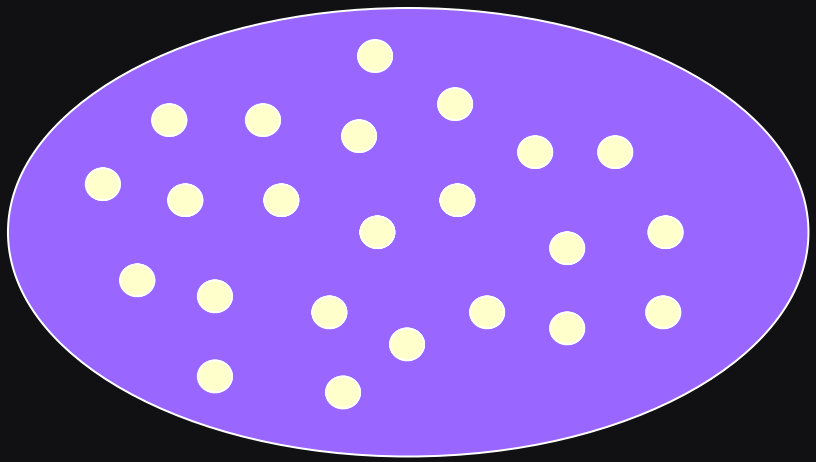
q_i's are integer multiples of δ

$$\left\| \sum_i X_i - \sum_i Y_i \right\|_{\mathrm{TV}} \leq O(\sqrt{\delta}) \quad \Rightarrow$$

ε - approximation in time $n^{O(1/\varepsilon^2)}$

approximation if we replace the X_i's by the Y_i's

in fact, an "oblivious" algorithm…



set S_ε of all unordered collections of mixed strategies which are integer multiples of ε^2

- sample an (anonymous) mixed profile from S_ε
- look at the game only to determine if the sampled strategies can be assigned to players to get an ε -approximate equilibrium (via a max-flow argument)
- expected running time $n^{O(1/\varepsilon^2)}$

Oblivious-ness Property: the set S_ε does not depend on the game we need to solve

is there a faster PTAS?

Theorem [Daskalakis '08]:

There is an oblivious PTAS with running time $poly(n) \cdot (1/\epsilon)^{O(1/\epsilon^2)}$

the underlying structural result…

Theorem [Daskalakis'08]: In every anonymous game there exists an ε -approximate Nash equilibrium in which

- either all players who mix play the same mixed strategy

- or, at most $1/\epsilon^3$ mix, and they choose mixed strategies which are integer multiples of ϵ^2

the corresponding symmetry…

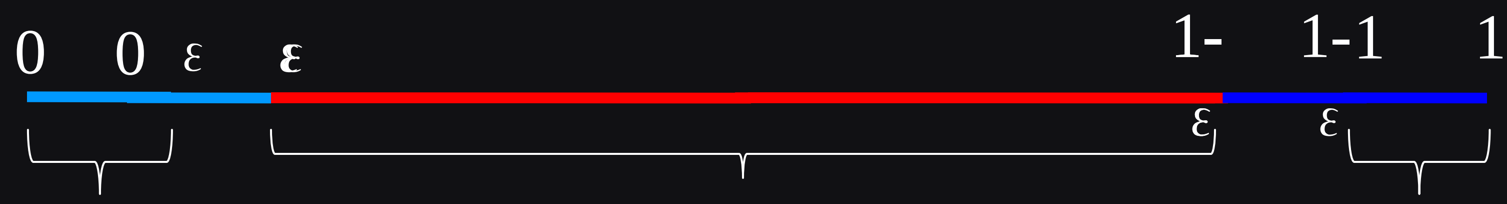
Lemma:

- The sum of $m \geq k^3$ indicators X_i with expectations in [1/k,1-1/k] is O(1/k)-close in total variation distance to a Binomial distribution with the same mean and variance

… i.e. close to a sum of indicators with the same expectation

[tightness of parameters by Berry-Esséen]

proof of structural result



round some of the X_i's falling here to 0 and some of them to ε so that the total mean is preserved to within ε

similarly

- if more than $1/\varepsilon^3$ X_i's are left here, appeal to previous slide (Binomial appx)

- o.w. use Dask. Pap. '07 (exists rounding into multiples of ε^2)

Final Result…

Theorem [Daskalakis'08]:

There is an oblivious PTAS with running time $poly(n) \cdot (1/\epsilon)^{O(1/\epsilon^2)}$

in fact this is essentially tight…

Theorem [Daskalakis, Papadimitriou '08]:

There is no oblivious PTAS with runtime better than

$$poly(n) \cdot 2^{\Omega(1/\epsilon^{1/3}))}$$

What about non-oblivious PTAS's?

Theorem [Daskalakis, Papadimitriou '08]:

There is a non-oblivious PTAS with running time

$$poly(n) \cdot (1/\epsilon)^{O(\log^2(1/\epsilon))}$$

the underlying probabilistic result [DP '08]:

If two sums of indicators have equal moments up to moment *k* then their total variation distance is $O(2^{-k})$.

now Kevin will continue our investigation of compact game representations, and their computational properties…

Equilibrium Computation in Compactly-Represented Games

Costis Daskalakis & Kevin Leyton-Brown

Part 2(b)

Overview

1. Congestion Games
2. Graphical Games
3. Action-Graph Games

Congestion Games

Congestion games [Rosenthal, 1973] are a restricted class of games with three key benefits:

- useful for modeling some important real-world settings
- attractive theoretical properties
- some positive computational results

Intuitively, they simplify the representation of a game by imposing constraints on the effects that a single agent's action can have on other agents' utilities.

Example

A computer network in which several users want to send large files at approximately the same time. What routes should they choose?

Definition

Intuitively, each player chooses some subset from a set of resources, and the cost of each resource depends on the number (but not identities) of other agents who select it.

Definition (Congestion game)

A congestion game is a tuple (N, R, A, c), where

- N is a set of n agents;
- R is a set of r *resources*;
- $A = A_1 \times \cdots \times A_n$, where $A_i \subseteq 2^R \setminus \{\emptyset\}$ is the set of *actions* for agent i; and
- $c = (c_1, \ldots, c_r)$, where $c_k : \mathbb{N} \mapsto \mathbb{R}$ is a *cost function* for resource $k \in R$.

From Cost Functions to Utility Functions

Definition ($\#(r,a)$)

Define $\#(r,a)$ as the number of players who took any action that involves resource r under action profile a.

Definition (Congestion game utility functions)

Given a pure-strategy profile $a = (a_i, a_{-i})$, let

$$u_i(a) = -\sum_{r \in R | r \in a_i} c_r(\#(r,a)).$$

- note: same utility function for all players
- negated, because cost functions are understood as penalties
  - however, the c_r functions may be negative
- anonymity property: players care about *how may* others use a given resource, but not about *which* others do so

Another Example: The Santa Fe Bar Problem

The cost functions don't have to increase monotonically in the number of agents using a resource.

Example (Santa Fe Bar Problem)

- People independently decide whether or not to go to a bar.
- The utility of attending increases with the number of others attending, up to the capacity of the bar.
- Then utility decreases because the bar gets too crowded.
- Deciding not to attend yields a baseline utility that does not depend on the actions of others.

A widely studied game.

- Famous for having no symmetric, pure-strategy equilibrium.
- Often studied in a repeated game context
- Generalized by so-called "minority games".

Pure Strategy Nash Equilibrium

The main motivation for congestion games was the following result:

Theorem (Rosenthal, 1973)

Every congestion game has a pure-strategy Nash equilibrium.

- This is a good thing, because pure-strategy Nash equilibria are more plausible than mixed-strategy Nash equilibria, and don't always exist.
- It also implies that the computational problem of finding an equilibrium in a congestion game is likely to be different
- Note that congestion games are exponentially more compact than their induced normal forms
    - if we're to find PSNE efficiently, we can't just check every action profile

Myopic Best Response

Myopic best response algorithm. It starts with an arbitrary action profile.

function $\textsc{MyopicBestResponse}$ (game G, action profile a) **returns** a
while *there exists an agent* i *for whom* a_i *is not a best response to* a_{-i} **do**
  $a'_i \leftarrow$ some best response by i to a_{-i}
  $a \leftarrow (a'_i, a_{-i})$
return a

- If it terminates, the algorithm returns a PSNE
- On general games, the algorithm doesn't terminate

Myopic Best Response

Myopic best response algorithm. It starts with an arbitrary action profile.

function $\textsc{MyopicBestResponse}$ (game G, action profile a) **returns** a
while *there exists an agent i for whom a_i is not a best response to a_{-i}*
do
  $a'_i \leftarrow$ some best response by i to a_{-i}
  $a \leftarrow (a'_i, a_{-i})$
return a

- If it terminates, the algorithm returns a PSNE
- On general games, the algorithm doesn't terminate

Theorem (Monderer & Shapley, 1996)

The $\textsc{MyopicBestResponse}$ *procedure is guaranteed to find a pure-strategy Nash equilibrium of a congestion game.*

This result depends on potential functions.

Potential Games

Definition (Potential game)

A game $G = (N, A, u)$ is a potential game if there exists some $P : A \mapsto \mathbb{R}$ such that, for all $i \in N$, all $a_{-i} \in A_{-i}$ and $a_i, a_i' \in A_i$, $u_i(a_i, a_{-i}) - u_i(a_i', a_{-i}) = P(a_i, a_{-i}) - P(a_i', a_{-i})$.

Potential Games

Definition (Potential game)

A game $G = (N, A, u)$ is a potential game if there exists some $P : A \mapsto \mathbb{R}$ such that, for all $i \in N$, all $a_{-i} \in A_{-i}$ and $a_i, a_i' \in A_i$, $u_i(a_i, a_{-i}) - u_i(a_i', a_{-i}) = P(a_i, a_{-i}) - P(a_i', a_{-i})$.

Theorem (Monderer & Shapley, 1996)

Every (finite) potential game has a pure-strategy Nash equilibrium.

Proof.

Let $a^* = \arg\max_{a \in A} P(a)$. Clearly for any other action profile a', $P(a^*) \geq P(a')$. Thus by the definition of a potential function, for any agent i who can change the action profile from a^* to a' by changing his own action, $u_i(a^*) \geq u_i(a')$.

Congestion Games have PSNE

Theorem (Rosenthal, 1973)

Every congestion game has a pure-strategy Nash equilibrium.

Proof.

Every congestion game has the following potential function:

$$P(a) = \sum_{r \in R} \sum_{j=1}^{\#(r,a)} c_r(j).$$

To show this, we must demonstrate that for any agent i and any action profiles (a_i, a_{-i}) and (a'_i, a_{-i}), the difference between the potential function evaluations at these action profiles is the same as i's difference in utility. This follows from a straightforward arithmetic argument; omitted.

Convergence of MyopicBestResponse

Theorem (Monderer & Shapley, 1996)

The MyopicBestResponse *procedure is guaranteed to find a pure-strategy Nash equilibrium of a congestion game.*

Proof.

It is sufficient to show that MyopicBestResponse finds a pure-strategy Nash equilibrium of any potential game. With every step of the while loop, $P(a)$ strictly increases, because by construction $u_i(a_i', a_{-i}) > u_i(a_i, a_{-i})$, and thus by the definition of a potential function $P(a_i', a_{-i}) > P(a_i, a_{-i})$. Since there are only a finite number of action profiles, the algorithm must terminate.

Analyzing the MyopicBestResponse result

Good news:

- it didn't require the cost functions to be monotonic
- it doesn't even require best response: it works with better response.

Analyzing the MyopicBestResponse result

Good news:

- it didn't require the cost functions to be monotonic
- it doesn't even require best response: it works with better response.

Bad news:

Theorem (Fabrikant, Papadimitriou & Talwar, 2004)

Finding a pure-strategy Nash equilibrium in a congestion game is PLS-complete.

- PLS-complete: as hard to find as any other object whose existence is guaranteed by a potential function argument.
    - e.g., as hard as finding a local minimum in a TSP using local search
- thus, we expect MyopicBestResponse to be inefficient in the worst case

Mixed Nash in Congestion Games

Not a problem that has received wide study. Nevertheless...

Theorem

Congestion games have polynomial type (as long as the action set for each player is explicitly listed). The EXPECTEDUTILITY *problem can be computed in polynomial time for congestion games, and such an algorithm can be translated to an straight-line program as required by the theorem stated earlier.*

Corollary

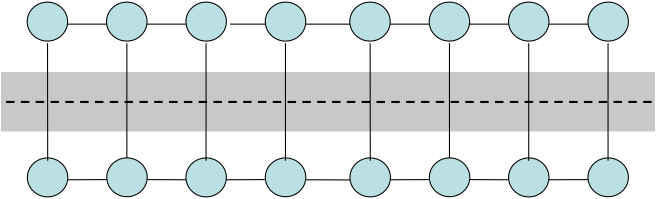
The problem of finding a Nash equilibrium of a congestion game is in PPAD. The problem of finding a correlated equilibrium of a congestion game is in P.

Overview

Graphical Games

- Graphical Games [Kearns et al., 2001] are a compact representation of normal-form games that use graphical models to capture the payoff independence structure of the game.
- Intuitively, a player's payoff matrix can be written compactly if his payoff is affected only by a subset of the other players.

Graphical Game Example



Example (Road game)

Consider n agents who have purchased pieces of land alongside a road. Each agent has to decide what to build on his land. His payoff depends on what he builds himself, what is built on the land to either side of his own, and what is built across the road.

Formal Definition

Definition (Neighborhood relation)

For a graph defined on a set of nodes N and edges E, for every $i \in N$ define the neighborhood relation $\nu : N \mapsto 2^N$ as $\nu(i) = \{i\} \cup \{j | (j,i) \in E\}$.

Definition (Graphical game)

A graphical game is a tuple (N, E, A, u), where:

- N is a set of n vertices, representing agents;
- E is a set of undirected edges connecting the nodes N;
- $A = A_1 \times \cdots \times A_n$, where A_i is the set of actions available to agent i; and
- $u = (u_1, \ldots, u_n)$, $u_i : A^{(i)} \mapsto \mathbb{R}$, where $A^{(i)} = \prod_{j \in \nu(i)} A_j$.

Representation Size

- An edge between two vertices $\Leftrightarrow$ the two agents are able to affect each other's payoffs
  - whenever two nodes i and j are *not* connected in the graph, agent i must always receive the same payoff under any action profiles (a_j, a_{-j}) and (a'_j, a_{-j}), $a_j, a'_j \in A_j$
- Graphical games can represent any game, but not always compactly
  - space complexity is exponential in the size of the largest $\nu(i)$
- In the road game:
  - the size of the largest $\nu(i)$ is 4, independent of the total number of agents
  - the representation requires space polynomial in n, while a normal-form representation requires space exponential in n

Computing CE and Mixed NE in Graphical Games

Theorem

Graphical games have polynomial type. The ExpectedUtility *problem can be computed in polynomial time for graphical games, and such an algorithm can be translated to an straight-line program.*

Corollary

The problem of finding a Nash equilibrium of a graphical game is in PPAD. The problem of finding a correlated equilibrium of a graphical game is in P.

Theorem (Daskalakis, Goldberg & Papadimitriou, 2006)

The problem of finding a Nash equilibrium of a graphical game is PPAD complete, even if the degree of the graph is at most 3, and there are only 2 strategies per player.

Computing Mixed NE in Graphical Games

- The way that graphical games capture payoff independence is similar to the way that Bayesian networks capture conditional independence in multivariate probability distributions.
- It should therefore be unsurprising that many computations on graphical games can be performed efficiently using algorithms similar to those proposed in the graphical models literature.

Theorem (Kearns, Littman & Singh, 2001)

When the graph (N, E) defines a tree, a message-passing algorithm can compute an ϵ-Nash equilibrium in time polynomial in $1/\epsilon$ and the size of the representation.

Theorem (Elkind, Goldberg & Goldberg, 2006)

When the graph (N, E) is a path or a cycle, a similar algorithm can find an exact equilibrium in polynomial time.

Computing PSNE in Graphical Games

Theorem (Gottlob, Greco & Scarcello, 2005)

Determining whether a pure-strategy equilibrium exists in a graphical game is NP complete.

- This result follows from seeing the problem as a CSP.

Computing PSNE in Graphical Games

Theorem (Gottlob, Greco & Scarcello, 2005)

Determining whether a pure-strategy equilibrium exists in a graphical game is NP complete.

- This result follows from seeing the problem as a CSP.
- The same insight can be leveraged to obtain results like:

Theorem (Gottlob, Greco & Scarcello, 2005; Daskalakis & Papadimitriou, 2006)

Deciding whether a graphical game has a pure Nash equilibrium is in P for all classes of games with bounded treewidth or hypertreewidth.

- It's possible to go even a bit further, to games with $O(\log n)$ treewidth

Overview

The Coffee Shop Problem

Google Local Canada

Web Images Groups News Local New! more »

category: Coffee Houses [Search]

e.g., "hotels in calgary" or "5000 dufferin street, toronto"

Search the map
Find businesses
Get directions

Local Print Email Link to this page

Search results for **category: Coffee Houses** in this map

A. Connoisseurs' **Coffee**
1075 Georgia Street West, Vancouver, BC V6E 3C9
(604) 683-1486

B. Melriches Coffeehouse
1244 Davie Street, Vancouver, BC V6E 1N3
(604) 689-5282

C. Hole In The Wall Cappuccino Bar
1030 Georgia Street West, Vancouver, BC V6E 2Y3
(604) 646-4653

D. Starbucks **Coffee** Co
1055 W Georgia, Vancouver, BC V5K 1A1
(604) 685-5882

E. Five Roses Bakery Cafe
1220 Bute Street, Vancouver, BC V6E 1Z8
(604) 669-8989

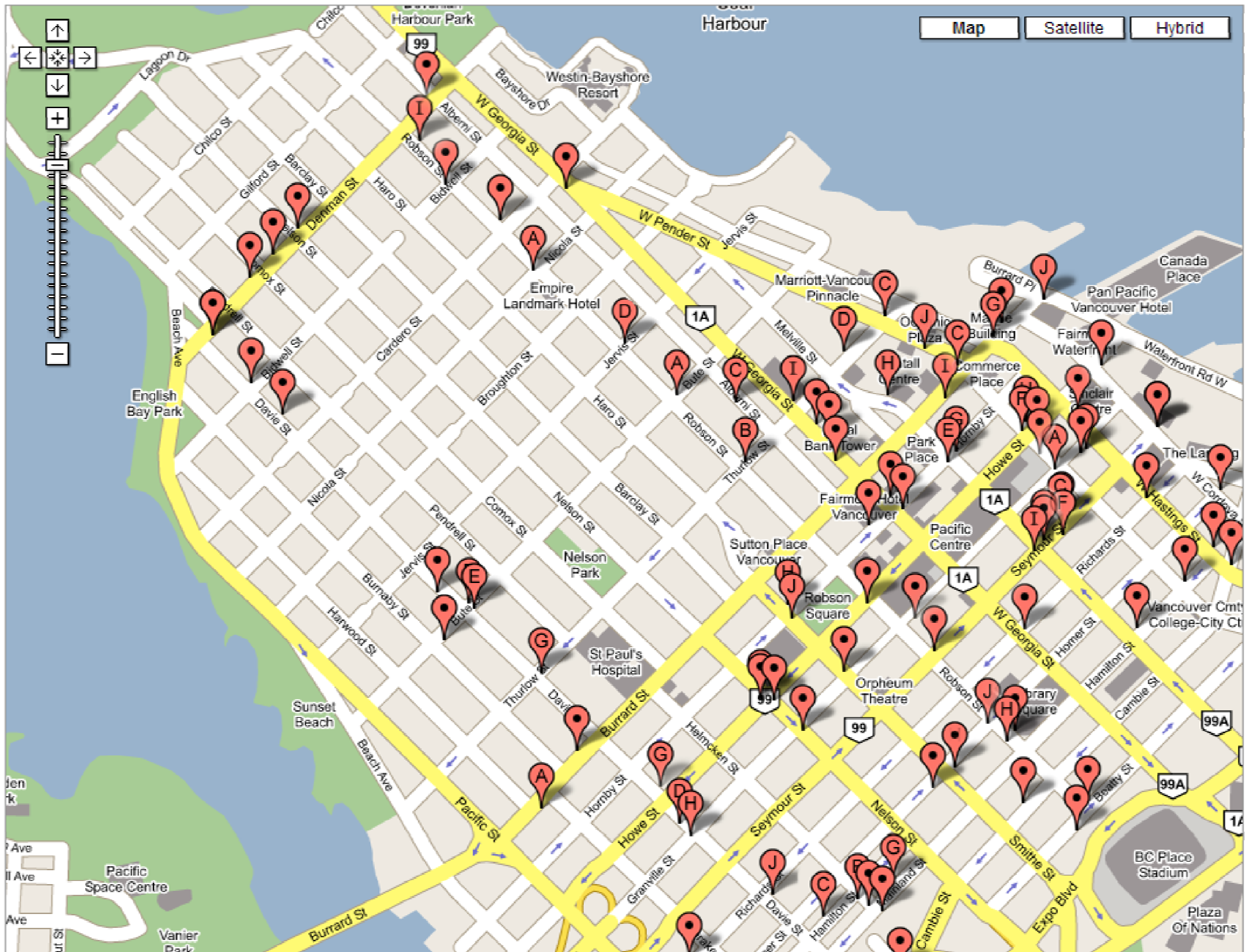
F. Starbucks **Coffee** Co
1095 Howe Street, Vancouver, BC V6Z 1P6
(604) 685-7083

G. Uptown Espresso
808 Nelson Street, Vancouver, BC V6Z 2H2
(604) 689-1920

H. Caffe Artigiano
763 Hornby Street, Vancouver, BC V6Z 1S2
(604) 696-9222

I. Skyline Expresso
900 Howe Street, Vancouver, BC V6Z 2M4
(604) 683-4234

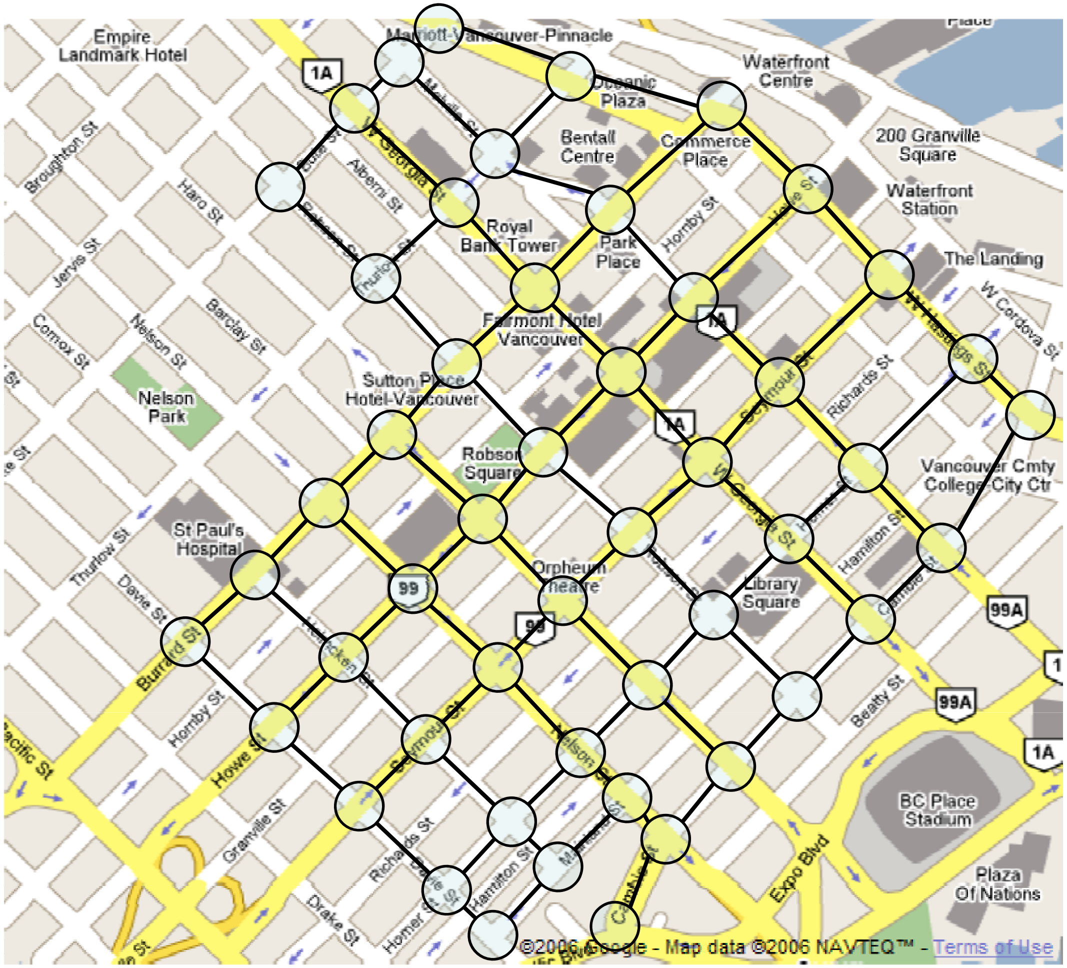
J. Farenheit Celsius **Coffee**
1225 Burrard Street, Vancouver, BC V6Z 1Z5
(604) 682-6675



Action-Graph Games

[Bhat & LB, 2004; Jiang, LB & Bhat, 2009]

- set of **players:** want to open coffee shops
- **actions:** choose a location for your shop, or choose not to enter the market
- **utility:** profitability of a location
  - some locations might have more customers, and so might be better *ex ante*
  - utility also depends on the number of other players who choose the same or an adjacent location



Formal Definitions

Definition 1 (action graph) *An* **action graph** *is a tuple* $(\mathcal{A}, E)$, *where* $\mathcal{A}$ *is a set of nodes corresponding to distinct actions and* E *is a set of directed edges.*

Let $A = (A_1, \ldots, A_n)$ be a ***set of actions*** available to each of n agents, with $\mathcal{A} = \bigcup_{i \in N} A_i$.

Definition 2 (configuration) *Given an action graph* $(\mathcal{A}, E)$ *and a set of action profiles* A, *a* **configuration** c *is a tuple of* $|\mathcal{A}|$ *non-negative integers, where the* j^{th} *element* $c[j]$ *is interpreted as the number of agents who chose the* j^{th} *action* $a_j \in \mathcal{A}$, *and where there exists some* $a \in A$ *that would give rise to* c. *Denote the set of all configurations as* C.

Formal Definitions

Definition 3 (neighborhood relation) *Given a graph having a set of nodes $\mathcal{A}$ and edges E, define the* **neighborhood relation** *as $\nu : \mathcal{A} \to 2^{\mathcal{A}}$, with $\nu(i) = \{j|(j,i) \in E\}$.*

Define a ***configuration over a node's neighborhood***, written as $c^{(\alpha)} \in C^{(\alpha)}$, as the elements of c that correspond to the actions $\nu(\alpha)$.

Definition 4 *An* **action-graph game (AGG)** *is a tuple (N, A, G, u), where:*

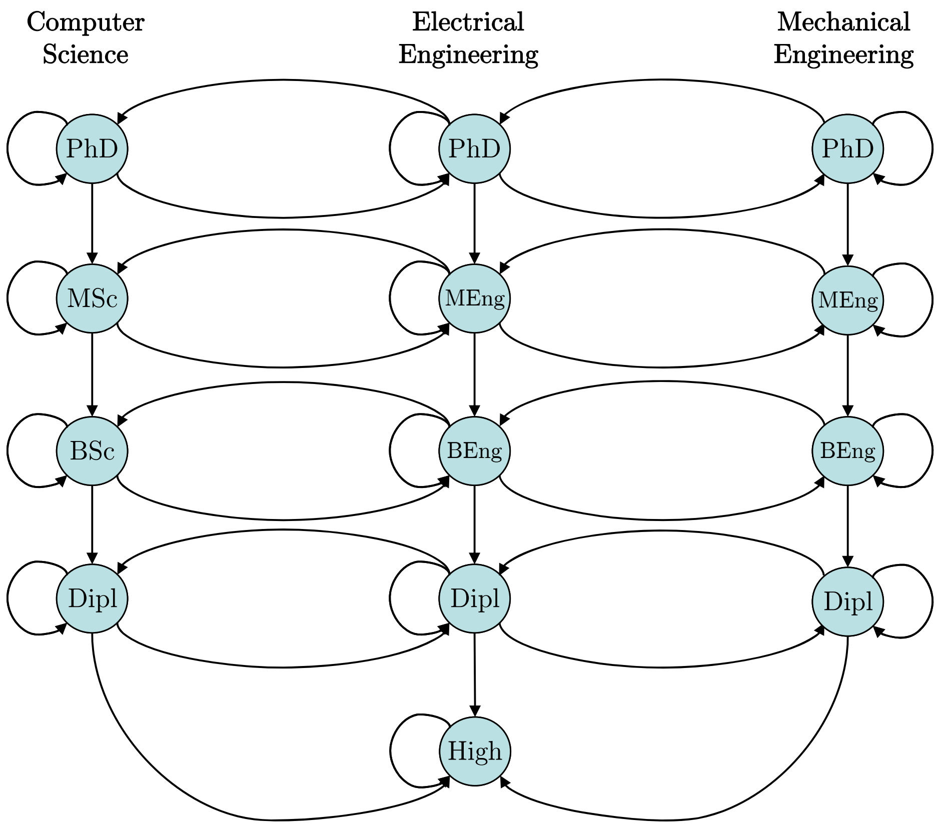
- *N is the set of agents;*
- *$A = (A_1, \ldots, A_n)$, where A_i is the set of actions available to agent i;*
- *$G = (\mathcal{A}, E)$ is an action graph, where $\mathcal{A} = \bigcup_{i \in N} A_i$ is the set of distinct actions;*
- $u = (u^1, \ldots, u^{|\mathcal{A}|})$, $u^\alpha : C^{(\alpha)} \to \mathbb{R}$.

The Job Market Problem

Each player chooses a level of training

Players' utilities are the sum of:

- a constant cost:
  - difficulty; tuition; foregone wages
- a variable reward, depending on:
  - How many jobs prefer workers with this training, and how desirable are the jobs?
  - How many other jobs are willing to take such workers as a second choice, and how good are these jobs?
    - Employers will take workers who are overqualified, but only by one degree.
    - They will also interchange similar degrees, but only at the same level.
  - How many other graduates want the same jobs?



Analyzing the AGG Representation

AGGs can represent any game.

Overall, AGGs are more compact than the normal form when the game exhibits either or both of the following properties:

1. **Context-Specific Independence:**
   - pairs of agents can choose actions that are not neighbors in the action graph

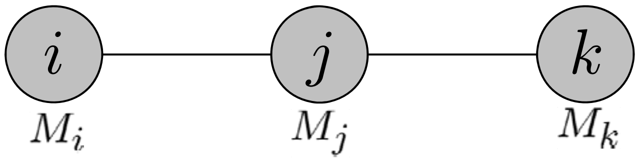
2. **Anonymity:**
   - multiple action profiles yield the same configuration

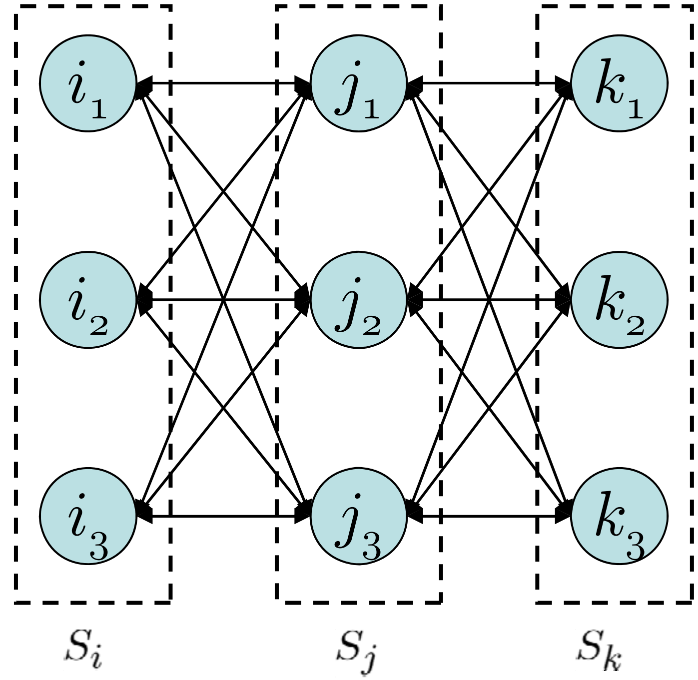
When max in-degree $\mathcal{I}$ is bounded by a constant:

– polynomial size: $O(|A_{\max}|n^{\mathcal{I}})$

– in contrast, size of normal form is $O(n|A_{\max}|^n)$

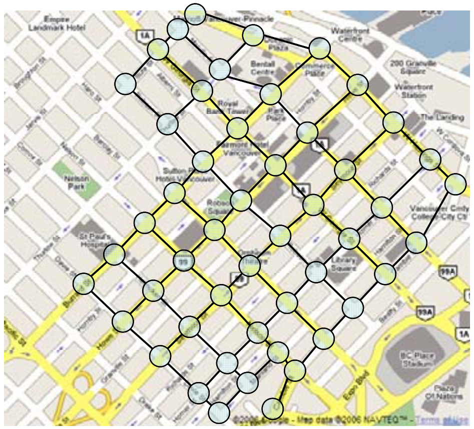
Graphical Games are Compact as AGGs



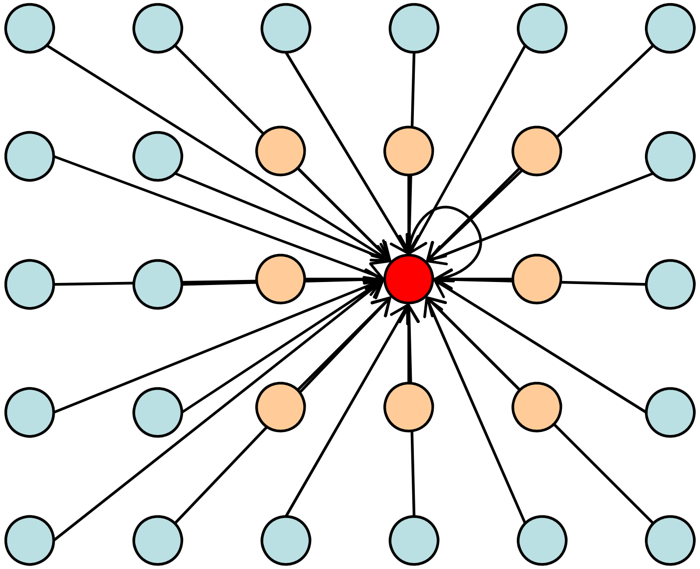


GG	AGG
Agent node	Action set box
Edge	Bipartite graphs between action sets
Local game matrix	Node utility function

The Coffee Shop Problem Revisited



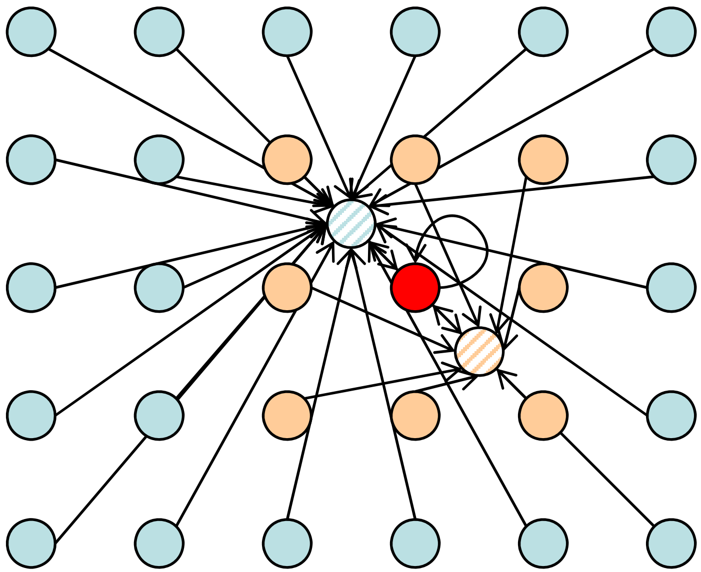
- What if utility also depends on total # shops?
- Now action graph has in-degree $|\mathcal{A}|$
  - NF & Graphical Game representations: $O(|\mathcal{A}|^N)$
  - AGG representation: $O(N^{|\mathcal{A}|})$
  - when $|\mathcal{A}|$ is held constant, the AGG representation is polynomial in N
    - but still doesn't effectively capture game structure
    - given i's action, his payoff depends only on 3 quantities!



6×5 Coffee Shop Problem: projected action graph at the red node

AGGFNs: Function Nodes

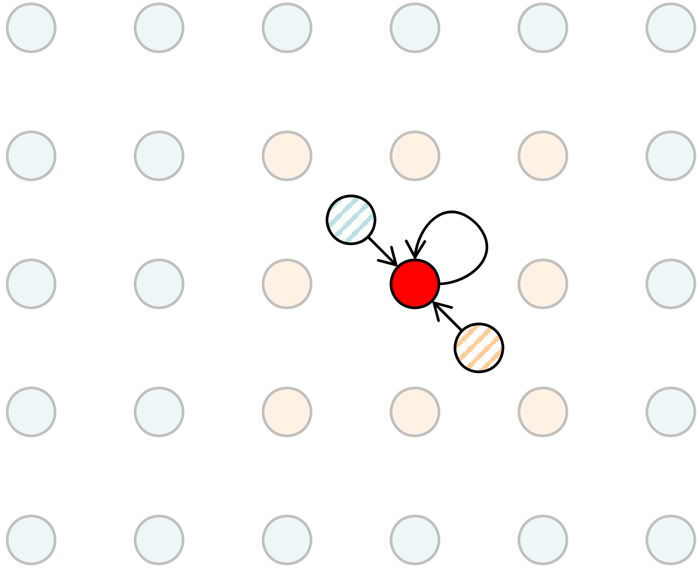
- To exploit this structure, introduce function nodes:
  - The "configuration" of a function node p is a (given) function of the configuration of its neighbors: $c[p] = f_p(c[\nu(p)])$
- Coffee-shop example: for each action node s, introduce:
  - a function node with adjacent actions as neighbors
    - $c[p'_s]$ = total number of shops in surrounding nodes
  - similarly, a function node with non-adjacent actions as neighbors



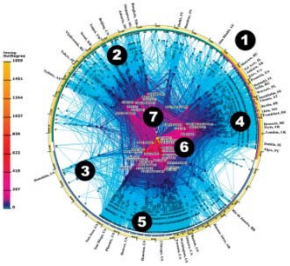
6×5 Coffee Shop Problem: function nodes for the red node

The Coffee Shop Problem

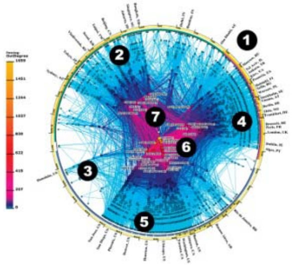
- Now the red node has only three incoming edges:
  - itself, the blue function node and the orange function node
  - so, the action-graph now has in-degree three
- Size of representation is now $O(N^3)$



6×5 Coffee Shop Problem: projected action graph at the red node

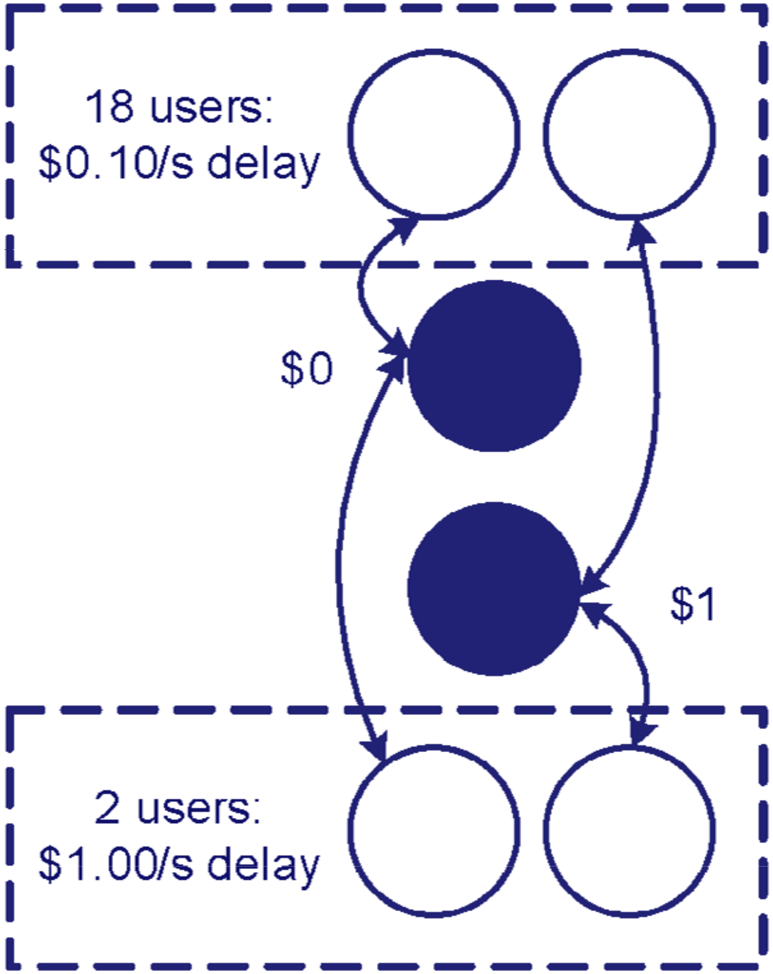


Example: Parallel Edges



Based on [Thompson, Jiang & LB, 2007]; inspired by [Odlyzko, 1998]

- Network with one source, one sink, two parallel edges
  - both edges offer identical speed
  - one is free, one costs \$1
  - latency is an additive function of the number of users on an edge
- Two classes of users
  - 18 users pay \$0.10/unit of delay
  - 2 users pay \$1.00/unit of delay
- Which edge should users choose?
- Example scales to longer paths
  - not a congestion game because of player-specific utility



Further Representational Results

[Jiang, LB & Bhat, 2009]

- Without loss of compactness, AGGs can also encode:
  - Symmetric games
  - Anonymous games (requires function nodes)

- One other extension to AGGs: explicit additive structure
- Enables compact encoding of still other game classes:
  - Congestion games
  - Polymatrix games
  - Local-Effect games

Conclusion: AGGs compactly encode all major compact classes of simultaneous-move games, and also many new games that are compact in none of these representations.

Computing with AGGs: Complexity

Theorem (Jiang & LB, 2006; independently proven in Daskalakis, Schoenebeck, Valiant & Valiant 2009): AGGs have polynomial type. The EXPECTEDUTILITY problem can be computed in polynomial time for AGGs, and such an algorithm can be translated to a straight-line program.

In AGGFNs, players are no longer guaranteed to affect c independently

- the computation is still polynomial when function nodes can be expressed using a commutative, associative operator

Corollary: The problem of finding a Nash equilibrium of an AGG is in PPAD. The problem of finding a correlated equilibrium of an AGG is in P.

Computing with AGGs: Complexity

Theorem (Daskalakis, Schoenebeck, Valiant & Valiant 2009): There exists a fully polynomial time approximation scheme for computing mixed Nash equilibria of AGGs with constant degree, constant treewidth and a constant number of distinct action sets (but unbounded number of actions).

If either of the latter conditions is relaxed without new restrictions being made, the problem becomes intractable.

Theorem (DSVV-09): It is PPAD–complete to compute a mixed Nash equilibrium in an AGG for which (1) the action graph is a tree and the number of distinct action sets is unconstrained, or (2) there are a constant number of distinct action sets and treewidth is unconstrained.

Computing Pure-Strategy Equilibrium

Theorem (Conitzer, personal communication, 2004; proven independently by Daskalakis et al., 2008): The problem of determining existence of a pure Nash equilibrium in an AGG is NP-complete, even when the AGG is symmetric and has maximum in-degree of three.

Theorem (Jiang & LB, 2007): For symmetric AGGs with bounded treewidth, existence of pure Nash equilibrium can be determined in polynomial time.

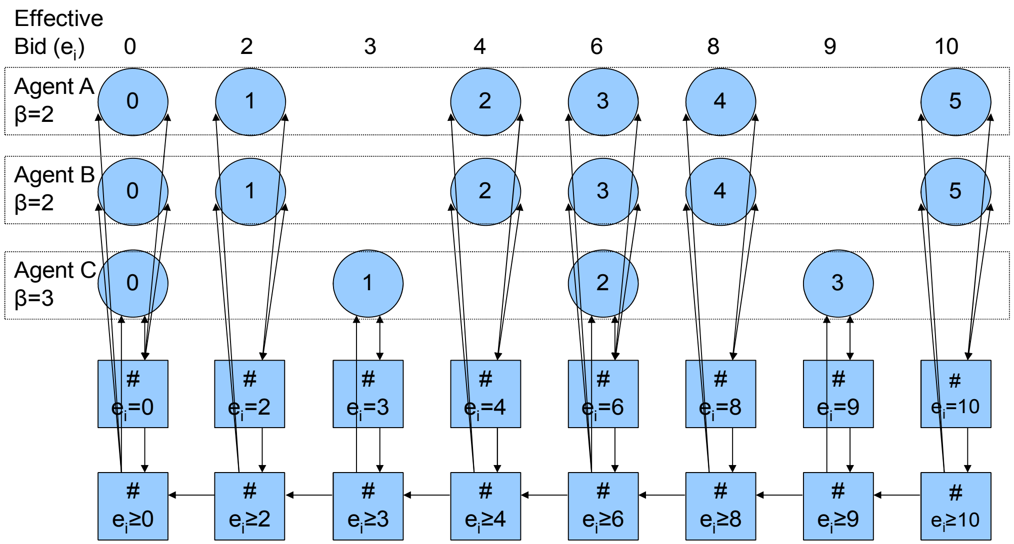
Generalizes earlier algorithms

- finding pure equilibria in graphical games
  [Gottlob, Greco, & Scarcello 2003; Daskalakis & Papadimitriou 2006]
- finding pure equilibria in simple congestion games
  [Ieong, McGrew, Nudelman, Shoham, & Sun 2005]

Sponsored Search Auctions

Brief preview of [Thompson & LB, ACM-EC 2009]

- Position auctions are used to sell $10Bs of keyword ads
- Some theoretical analysis, but based on strong assumptions
  - Unknown how different auctions compare in more general settings
- Idea: analyze the auctions computationally
  - Main hurdle: ad auction games are large; infeasible as normal form

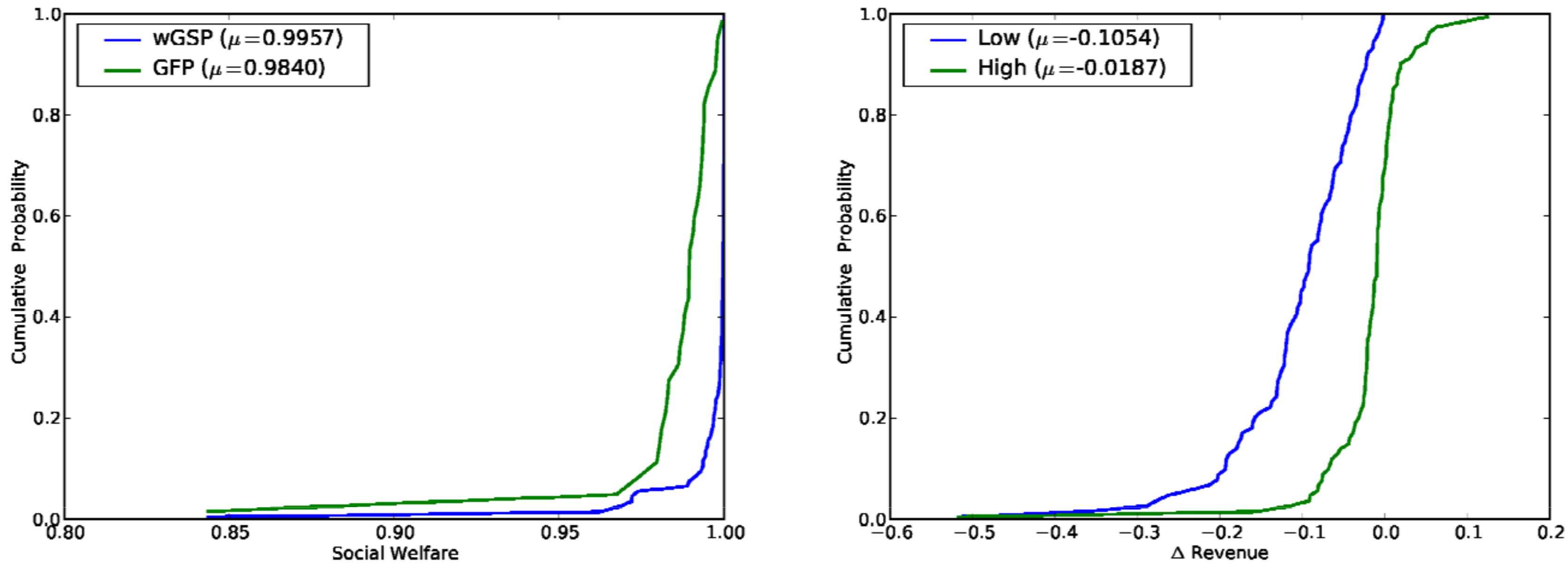


AGG representation of a Weighted, Generalized First-Price (GFP) Auction

Sponsored Search Auctions

Brief preview of [Thompson & LB, ACM-EC 2009]

- Position auctions are used to sell $10Bs of keyword ads
- Some theoretical analysis, but based on strong assumptions
  - Unknown how different auctions compare in more general settings
- Idea: analyze the auctions computationally
  - Main hurdle: ad auction games are large; infeasible as normal form

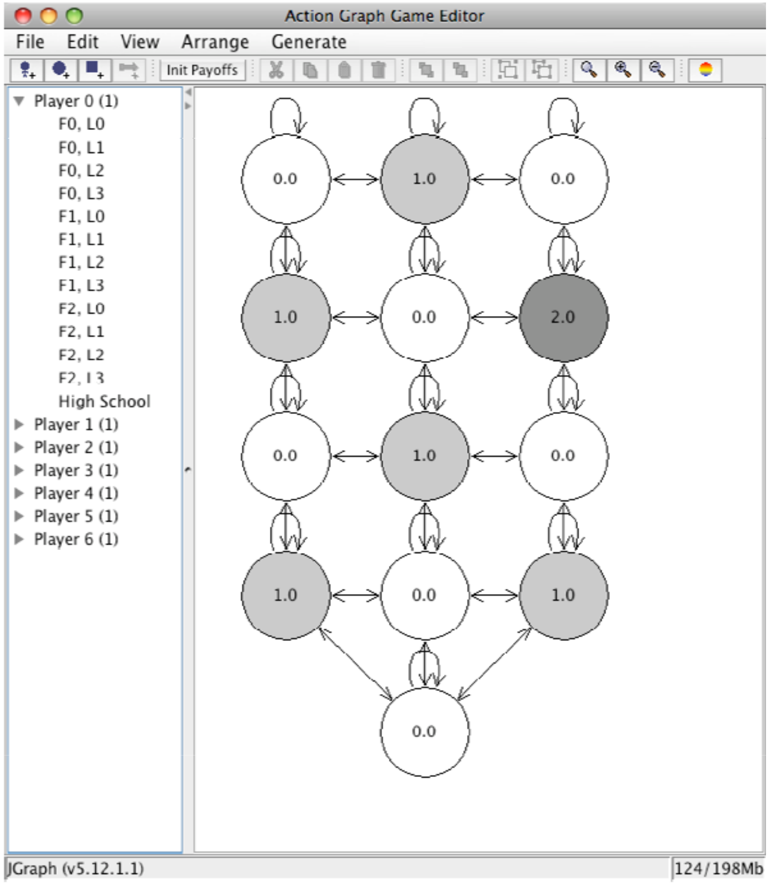


Social welfare and revenue of EOS auction model

Free Software Tools for AGGs

Based on [Bargiacchi, Jiang & LB, ongoing work]

- Goal: make it easier for other researchers to use AGGs
- Equilibrium computation algorithms:
  - Govindan-Wilson
  - Simplicial Subdivision
- GAMUT
  - extended to support AGGs
- Action Graph Game Editor:
  - creates AGGs graphically
  - facilitates entry of utility fns
  - supports "player classes"
  - auto creates game generators
  - visualizes equilibria on the action graph



Overall Conclusions

- Equilibrium computation is a hot topic lately
  - by now, the general complexity picture is fairly clear
- Compact representations are a fruitful area of study
  - necessary for modeling large-scale game-theoretic interactions
- There's lots to do, both in theoretical and applied veins
  - theoretical: only scratched the surface of restricted subclasses of games, and corresponding algorithmic and complexity results
  - both: extend our existing representations to make them more useful
  - applied: now that we have practical techniques for representing and reasoning with large games, see what practical problems we can solve
- We've focused on simultaneous-move, perfect-information games
  - the most fundamental, both representationally and computationally
  - to some extent, computational ideas carry over, both to incomplete information and to sequential moves
  - lots of interesting work on those problems that we haven't discussed
    - e.g., sequence form; algorithms for finding equilibria in huge extensive form games (motivated especially by poker); MAIDs, TAGGs