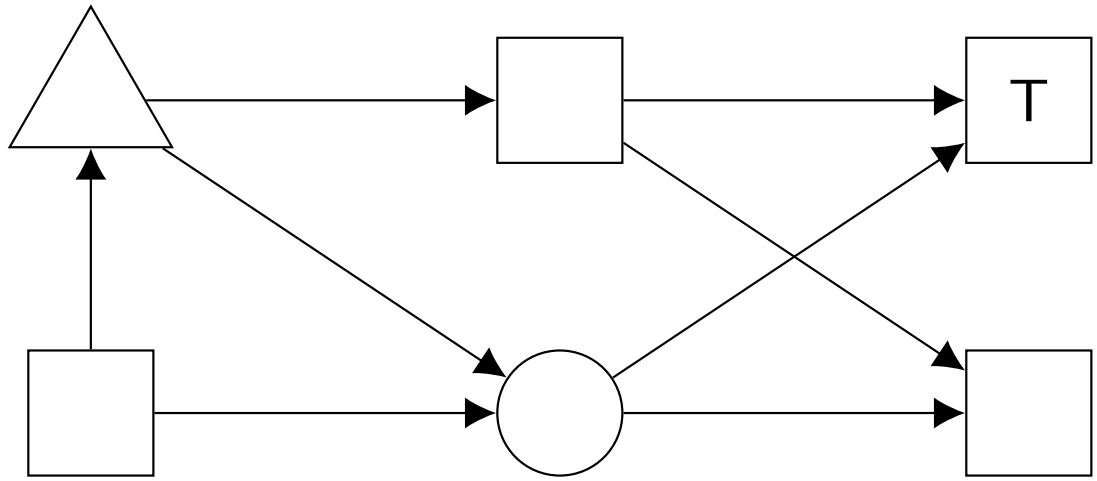
A Tour of the Complexity Classes Between P and NP

John Fearnley

University of Liverpool

Joint work with Spencer Gordon, Ruta Mehta, Rahul Savani

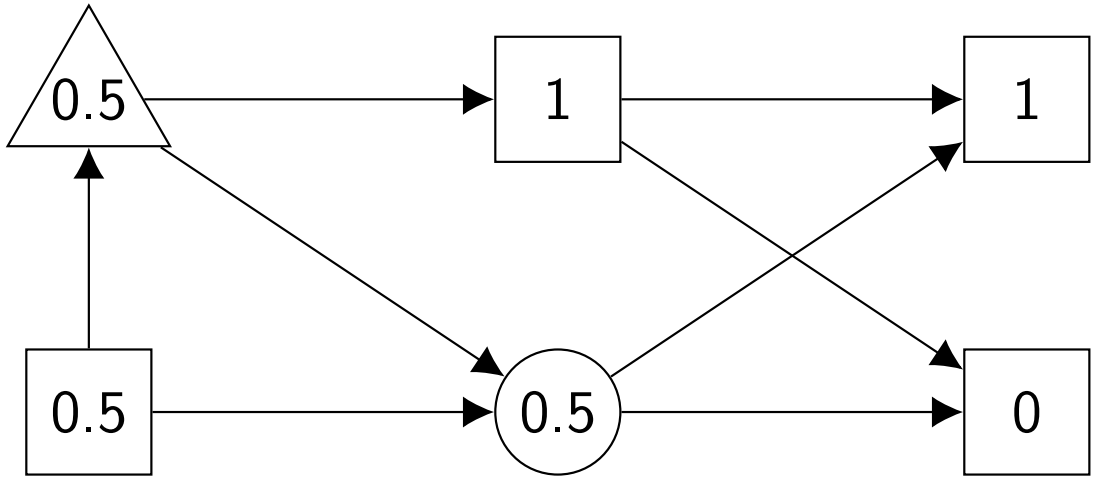
Simple stochastic games



A two player game

- Maximizer (box) wants to reach T
- Minimizer (triangle) who wants to avoid T
- Nature (circle) plays uniformly at random

Simple stochastic games

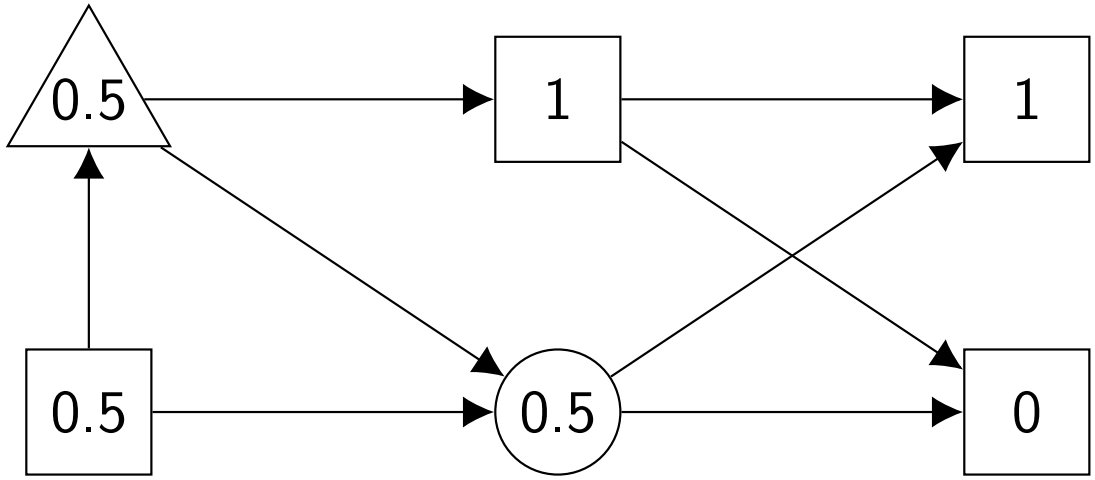


Value of a vertex:

- The largest probability of winning that max can ensure
- The smallest probability of winning that min can ensure

Computational Problem: find the value of each vertex

Simple stochastic games

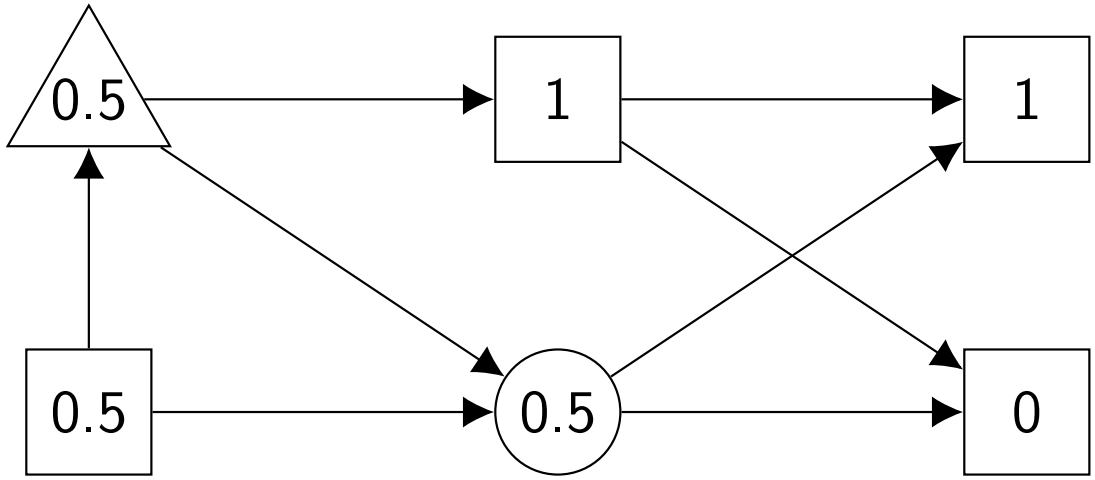


Is the problem

- **Easy?** Does it have a polynomial time algorithm?
- **Hard?** Perhaps no such algorithm exists

This is currently unresolved

Simple stochastic games



The problem lies in NP ∩ co-NP

- So it is unlikely to be NP-hard

But there are a lot of **NP-intermediate** classes...

Simple stochastic games

Solving a simple-stochastic game lies in

$$\text{NP} \cap \text{co-NP} \cap \text{UP} \cap \text{co-UP} \cap \text{TFNP} \cap$$

$$\text{PPP} \cap \text{PPA} \cap \text{PPAD} \cap \text{PLS} \cap$$

$$\text{CLS} \cap \text{EOPL} \cap \text{UEOPL}$$

Simple stochastic games

Solving a simple-stochastic game lies in

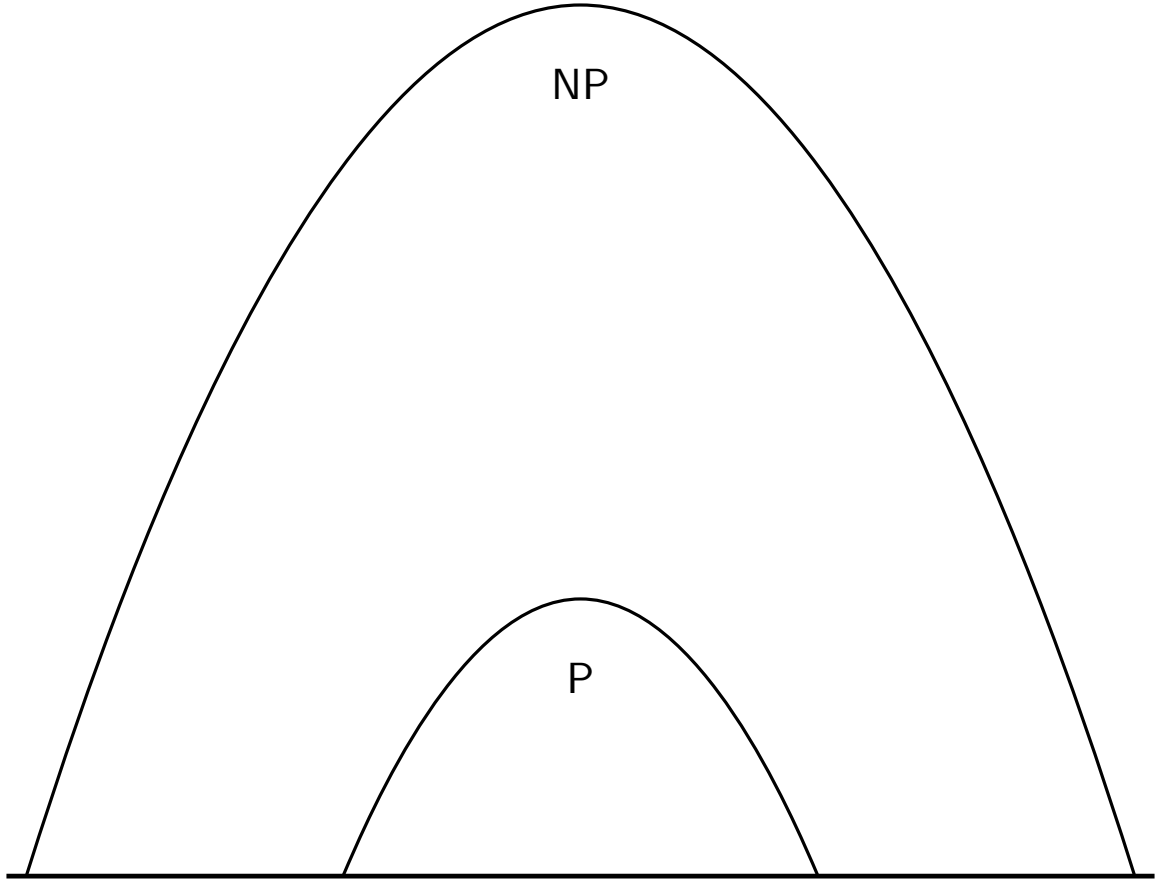
$$\text{NP} \cap \text{co-NP} \cap \text{UP} \cap \text{co-UP} \cap \textbf{TFNP} \cap$$

$$\text{PPP} \cap \text{PPA} \cap \textbf{PPAD} \cap \textbf{PLS} \cap$$

$$\textbf{CLS} \cap \textbf{EOPL} \cap \textbf{UEOPL}$$

This talk: whare are these complexity classes?

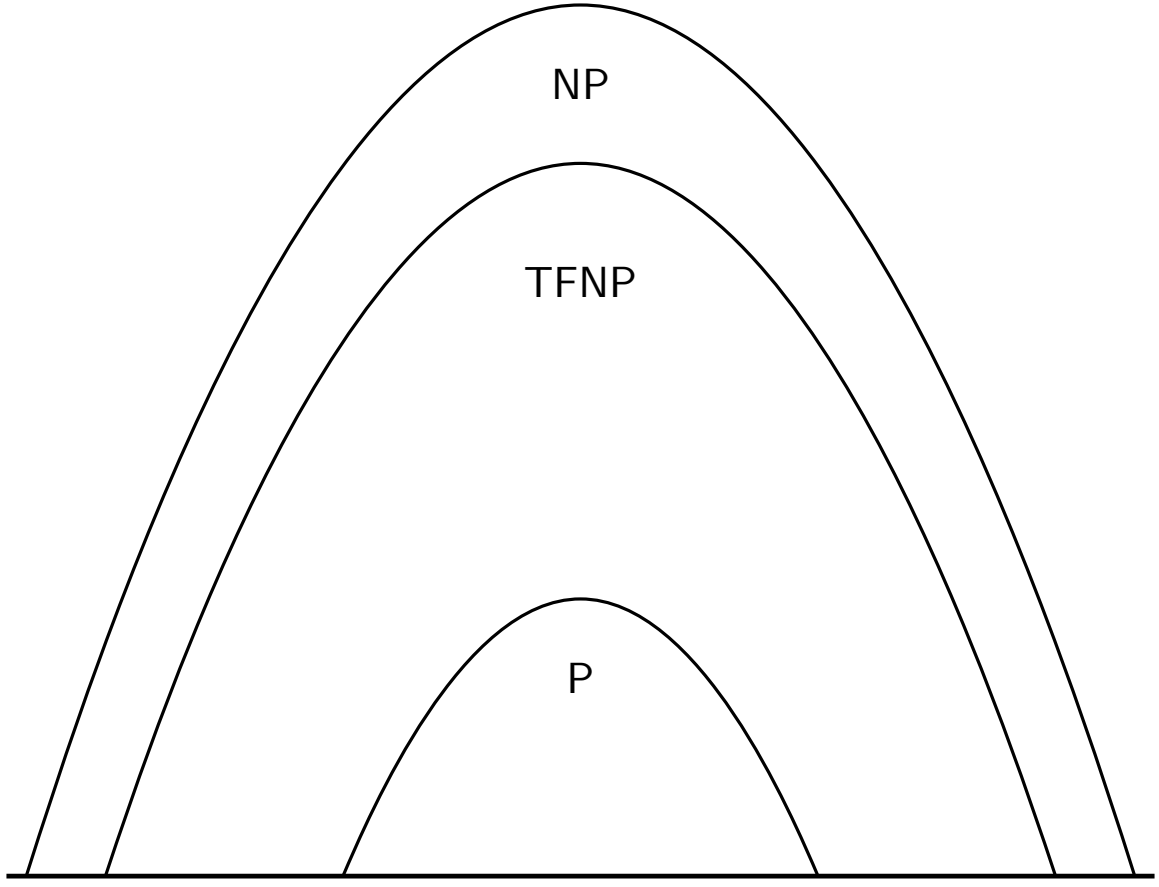
Complexity classes between P and NP



There are many problems that lie **between** P and NP

- Factoring, graph isomorphism, computing Nash equilibria, local max cut, simple-stochastic games, ...

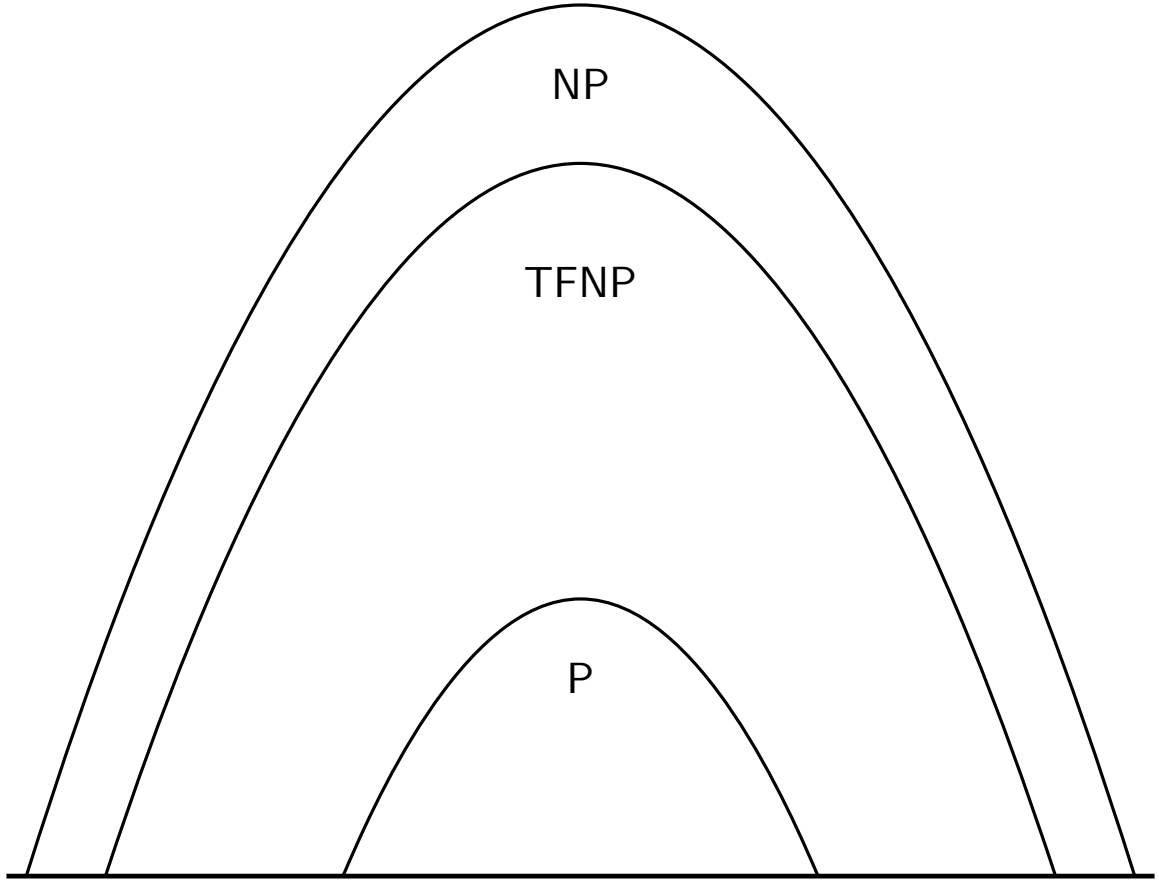
Complexity classes between P and NP



FNP is the class of **function** problems in NP

- Given polynomial time computable relation R and value x
- Find y such that $(x, y) \in R$

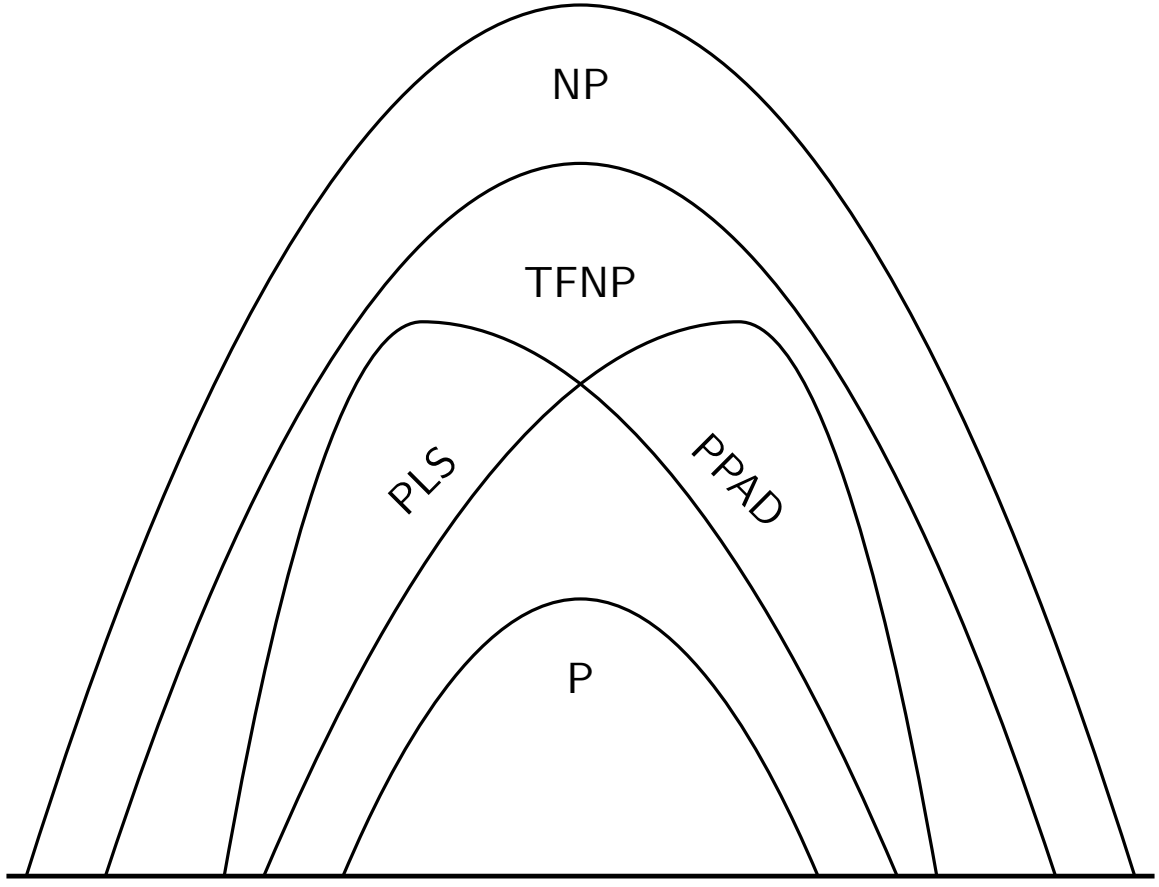
Complexity classes between P and NP



TFNP is the subclass of problems that **always** have solutions

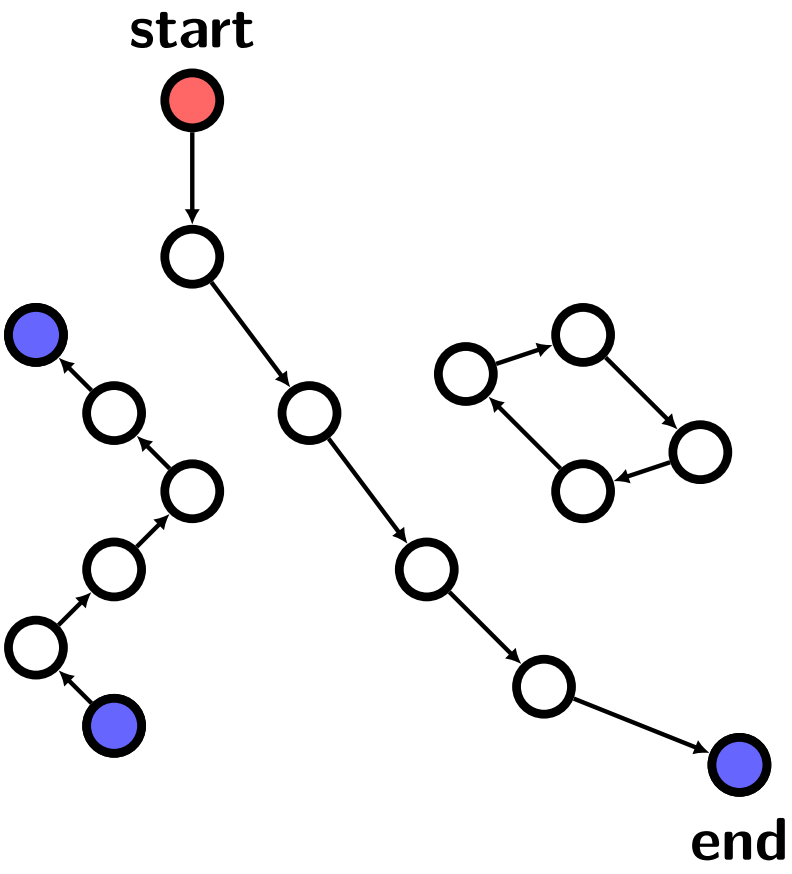
- Contains factoring, Nash equilibria, local max cut, simple-stochastic games, ...

Complexity classes between P and NP



PPAD and PLS are two **subclasses** of TFNP

PPAD (Papadimitriou 1994)

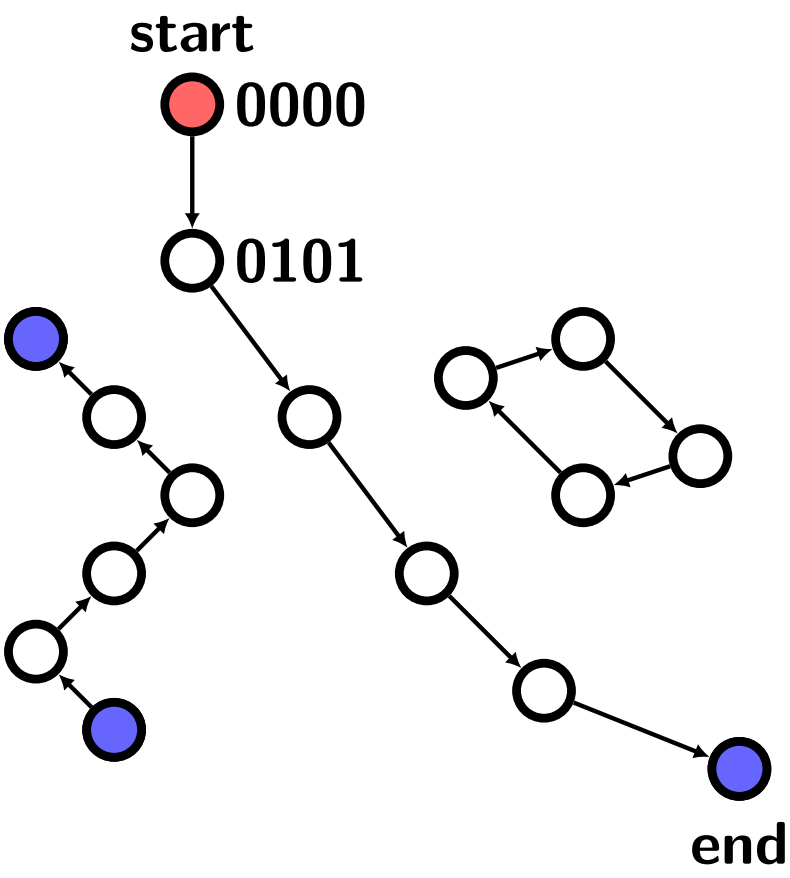


End-of-the-Line:

Given a graph G of in/out degree at most 1 and a **source start** vertex

find another vertex of degree 1

PPAD (Papadimitriou 1994)



Catch:

The graph is **exponentially large**

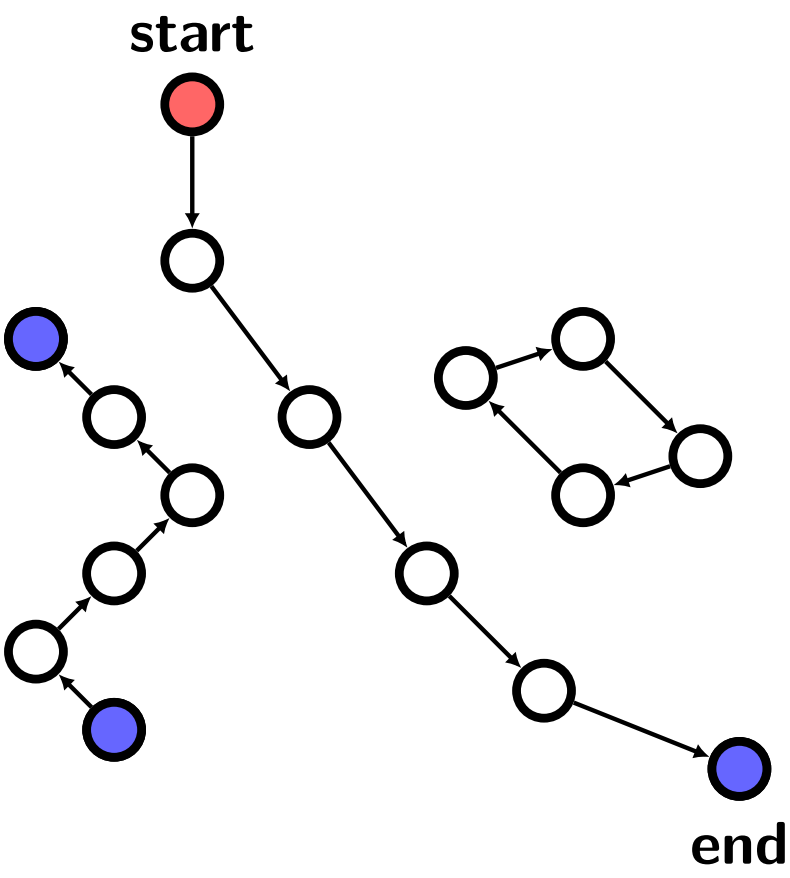
It is defined by

- A circuit S that gives a successor
- A circuit P that gives a predecessor

$$S(0000) = 0101$$

$$P(0101) = 0000$$

PPAD (Papadimitriou 1994)

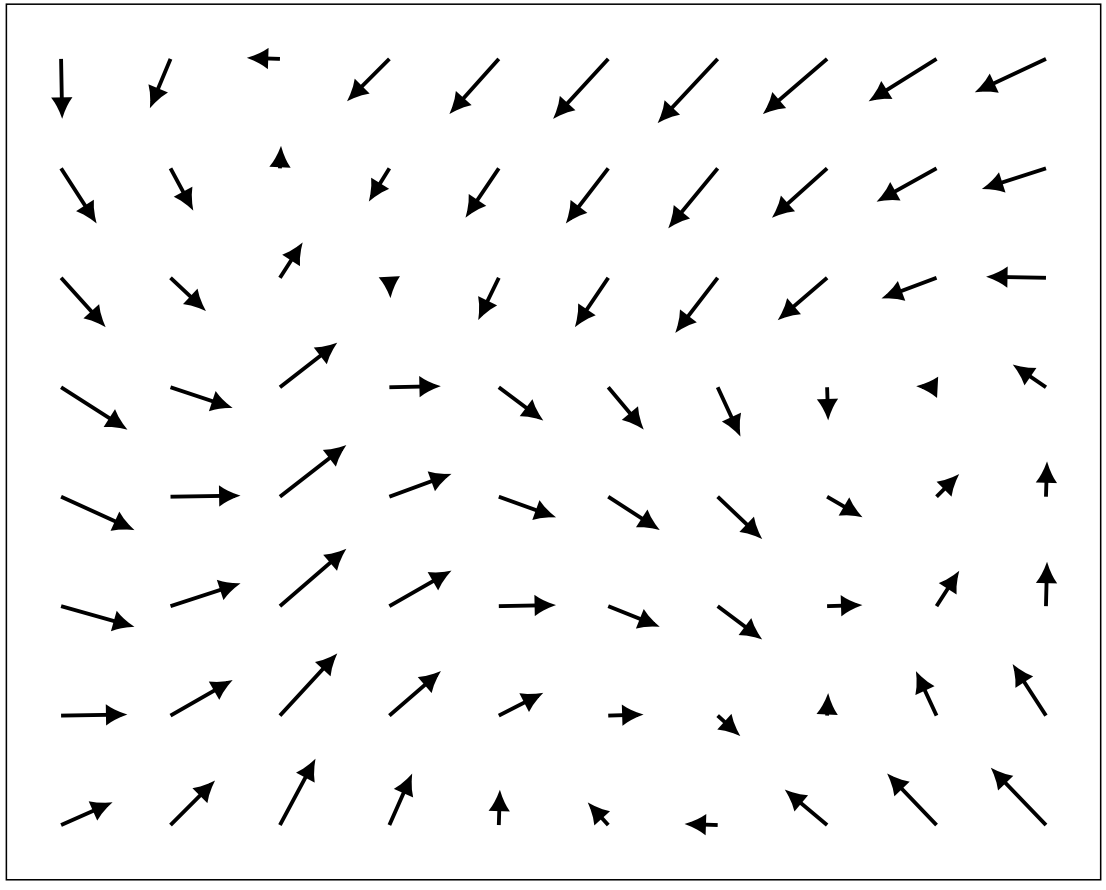


A problem is in PPAD if it can be reduced to EOTL

A problem is **PPAD-hard** if EOTL can be reduced to it

- Convincing evidence of hardness?

Brouwer: A PPAD-complete problem



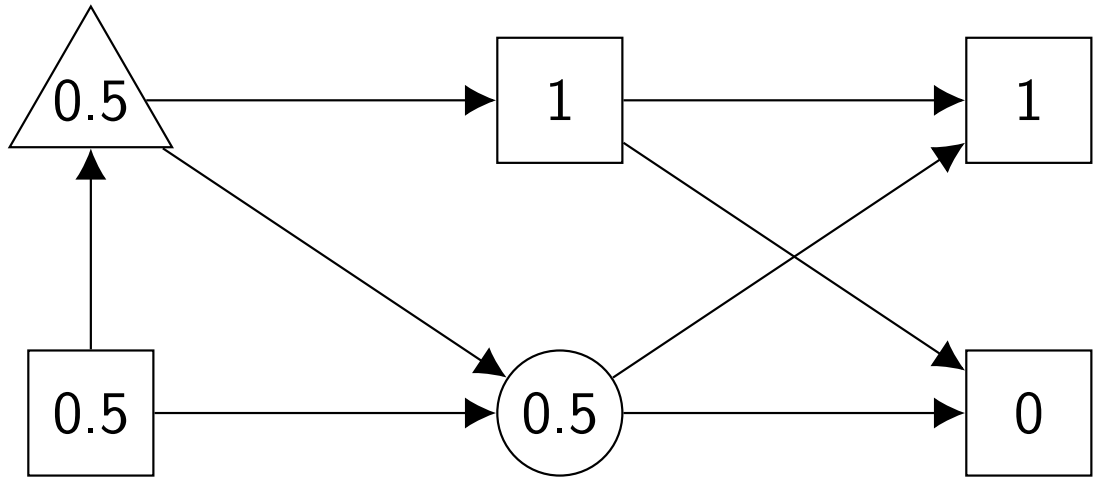
Given a continuous function $f : [0,1]^n \to [0,1]^n$

- find a **fixpoint**: a point x such that $f(x) = x$

PPAD-complete problems

- computing mixed equilibria in games
- computing Brouwer fixed points
- computing market equilibria

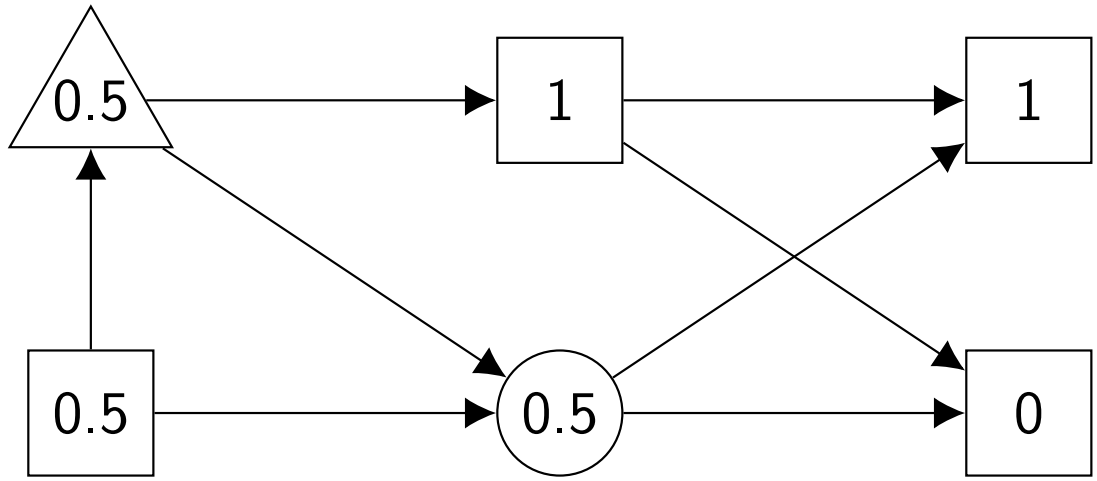
Simple stochastic games are in PPAD



The vertex values satisfy **optimality equations**

$$f(v) = \begin{cases} 1 & \text{if v is the target} \\ 0 & \text{if v is any other sink} \\ \max(u, w) & \text{if v is a max vertex with successors u and w} \\ \min(u, w) & \text{if v is a min vertex with successors u and w} \\ (u + w)/2 & \text{if v is a random vertex with successors u and w} \end{cases}$$

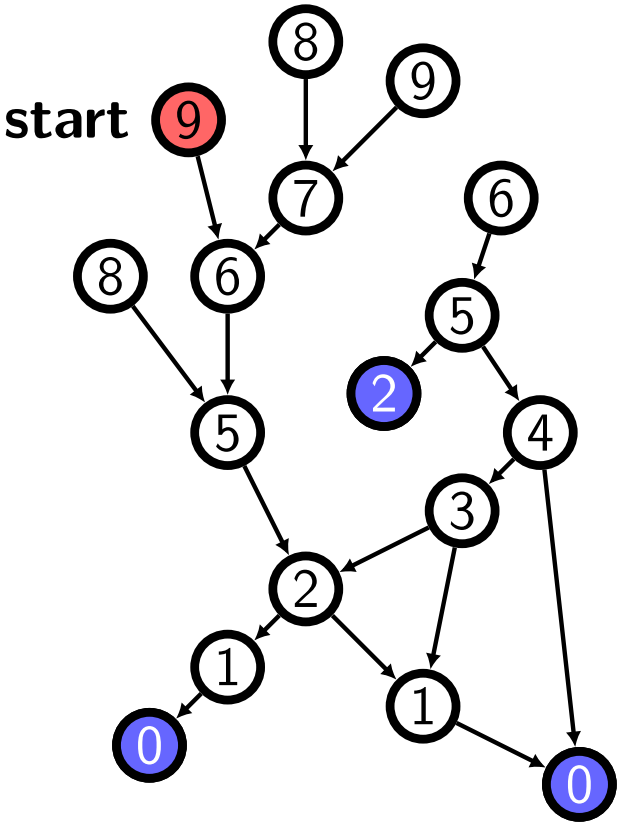
Simple stochastic games are in PPAD



f: $[0,1]^n \to [0,1]^n$ is a Brouwer function

When the game is **stopping**, the fixpoint of f gives the values for each vertex (Condon 1992)

Polynomial Local Search (PLS)



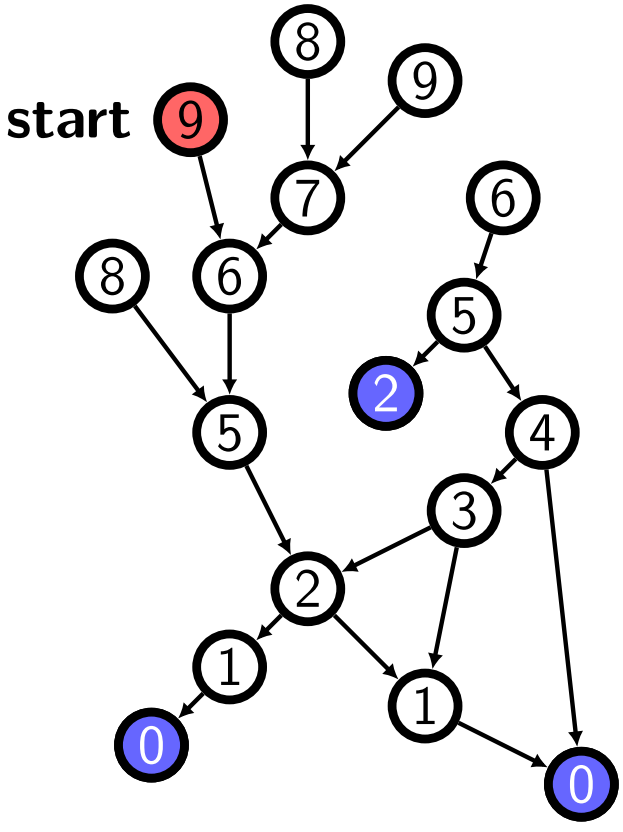
Given

- a DAG
- a starting vertex

Find

- a sink vertex

Polynomial Local Search (PLS)



Catch:

The graph is **exponentially large**

Defined by

- A circuit S giving the successor vertices
- A circuit p giving a **potential**

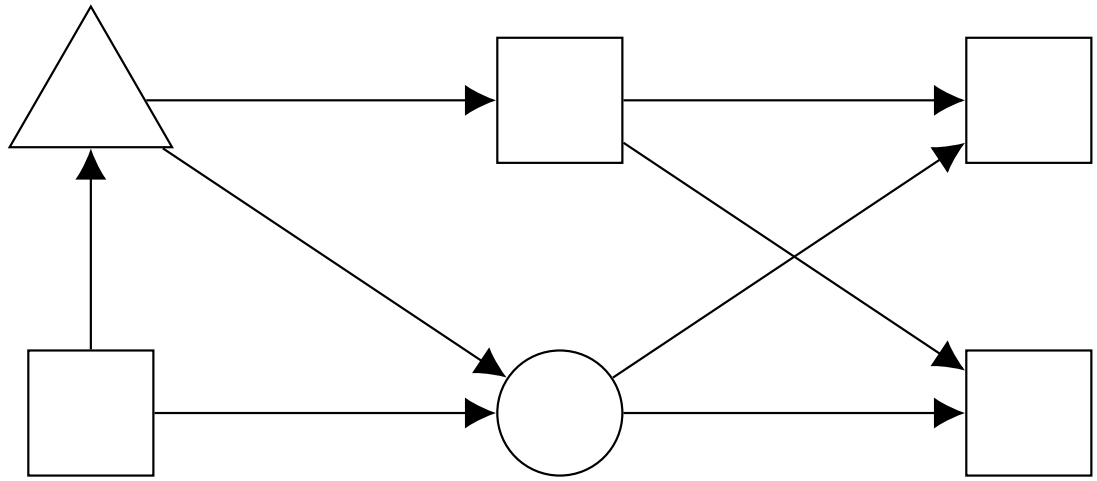
Every edge decreases the potential

$$p(S(v)) < p(v)$$

PLS-complete problems:

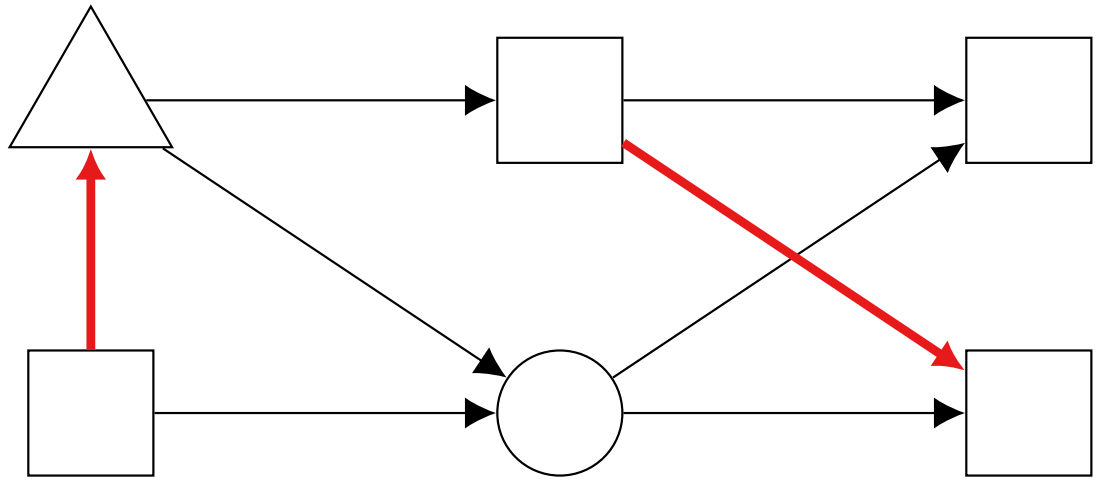
- local max cut
- computing pure equilibria in congestion games
- computing stable outcomes in hedonic games

Simple stochastic games are in PLS



A **strategy improvement** algorithm puts SSGs in PLS

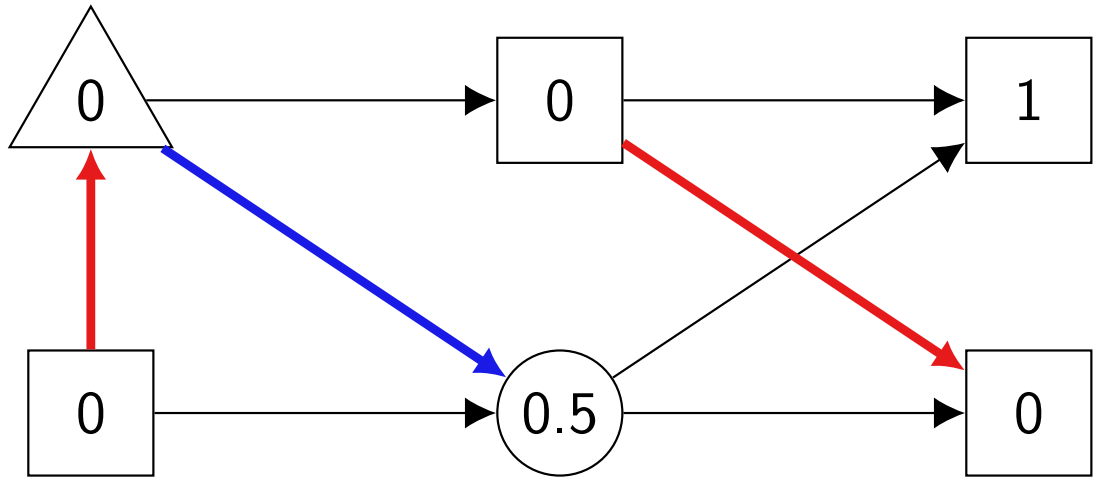
Simple stochastic games are in PLS



A **strategy improvement** algorithm puts SSGs in PLS

1. Maximizer picks a positional strategy

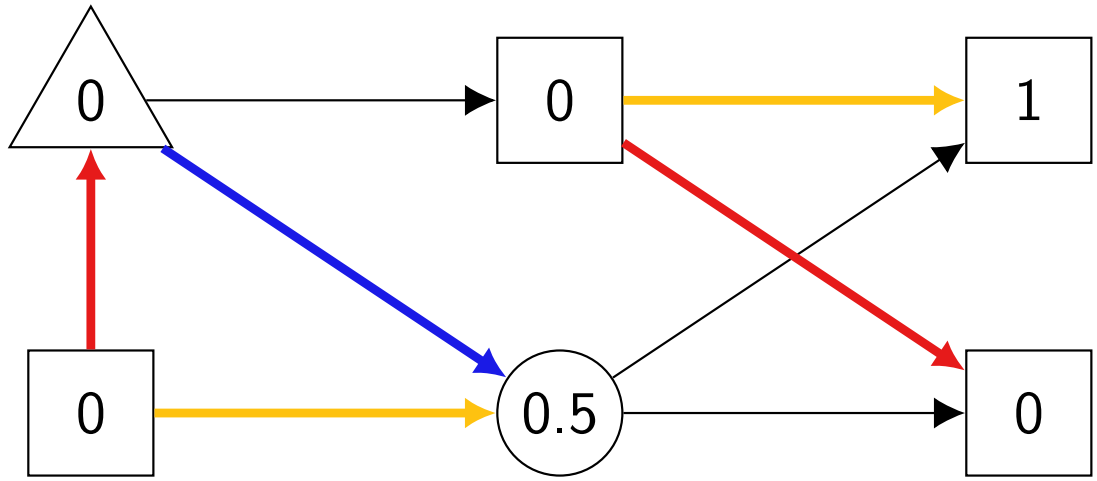
Simple stochastic games are in PLS



A **strategy improvement** algorithm puts SSGs in PLS

1. Maximizer picks a positional strategy
2. Compute a best response for the Minimizer

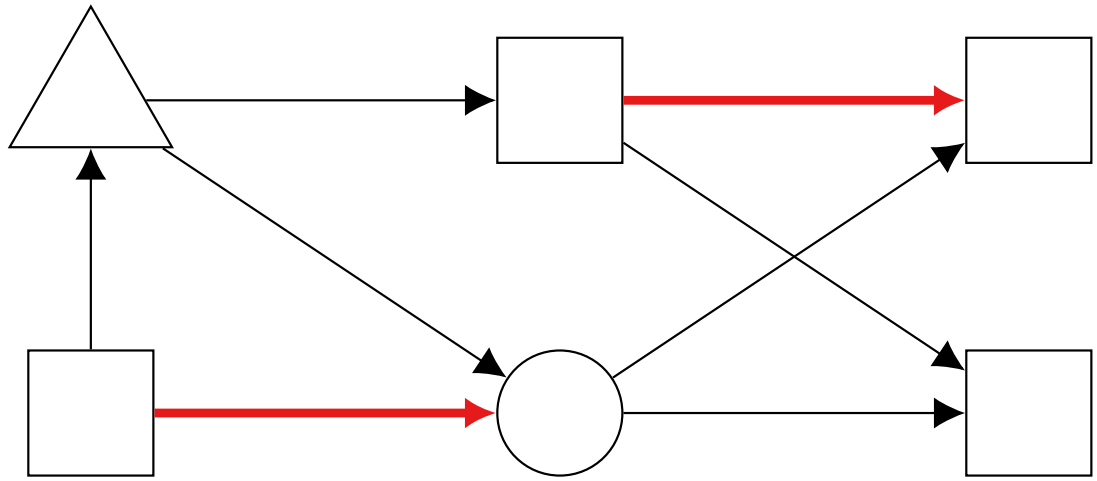
Simple stochastic games are in PLS



A **strategy improvement** algorithm puts SSGs in PLS

1. Maximizer picks a positional strategy
2. Compute a best response for the Minimizer
3. Determine switchable edges for Maximizer

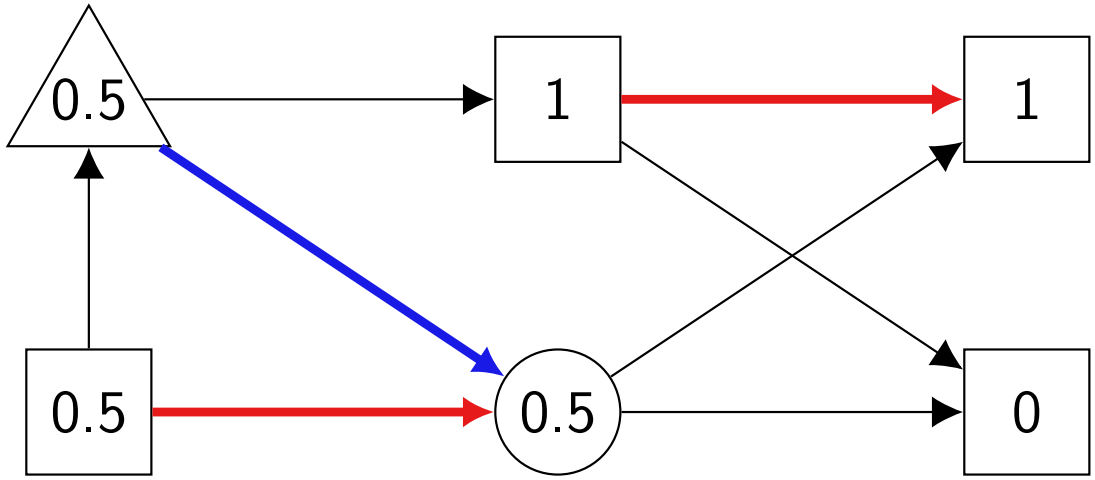
Simple stochastic games are in PLS



A **strategy improvement** algorithm puts SSGs in PLS

1. Maximizer picks a positional strategy
2. Compute a best response for the Minimizer
3. Determine switchable edges for Maximizer
4. Switch some switchable edges

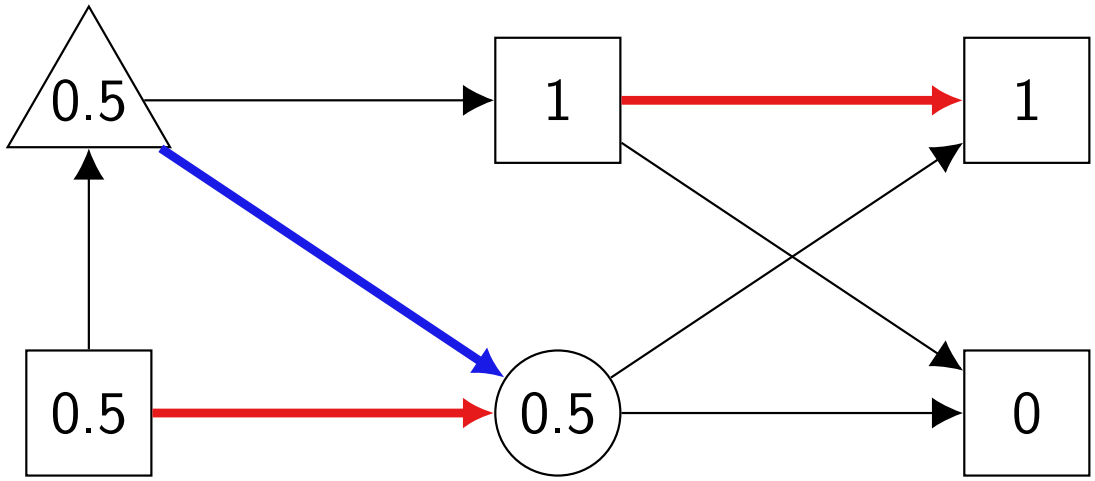
Simple stochastic games are in PLS



A **strategy improvement** algorithm puts SSGs in PLS

1. Maximizer picks a positional strategy
2. Compute a best response for the Minimizer
3. Determine switchable edges for Maximizer
4. Switch some switchable edges
5. Compute a new best response

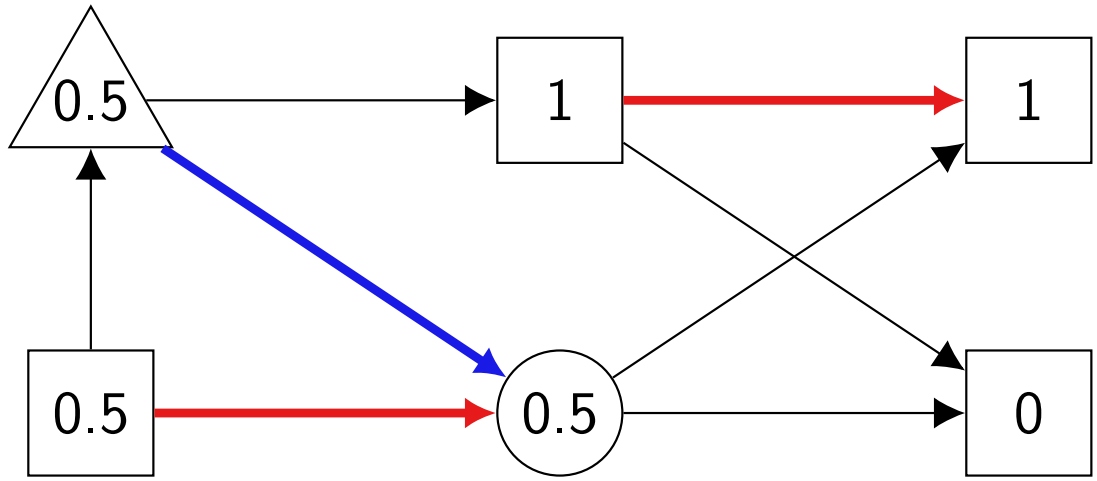
Simple stochastic games are in PLS



After switching any subset of switchable edges

- No vertex decreases in value
- At least one vertex **strictly increases** in value

Simple stochastic games are in PLS



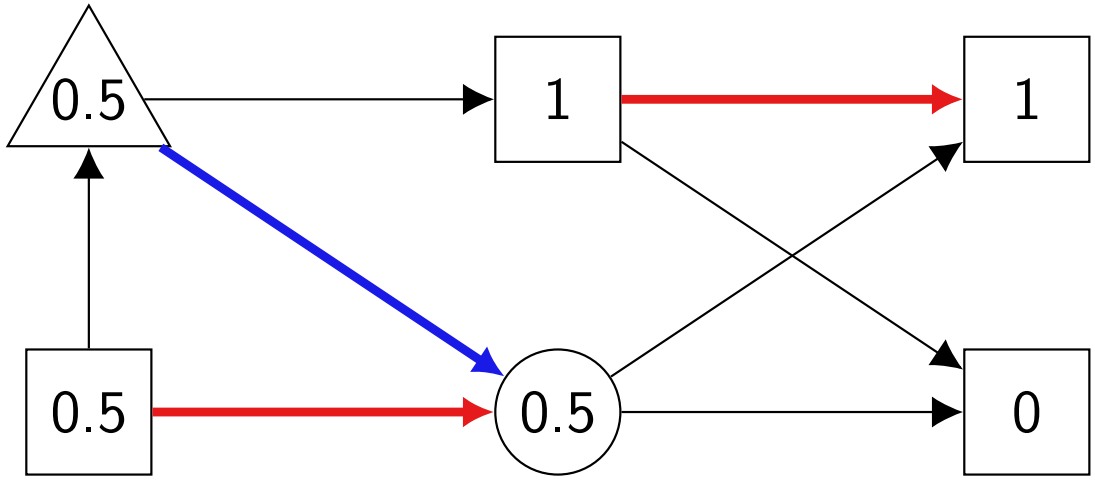
After switching any subset of switchable edges

- No vertex decreases in value
- At least one vertex **strictly increases** in value

If a Maximizer strategy has **no switchable edges**

- We have found the value of all vertices

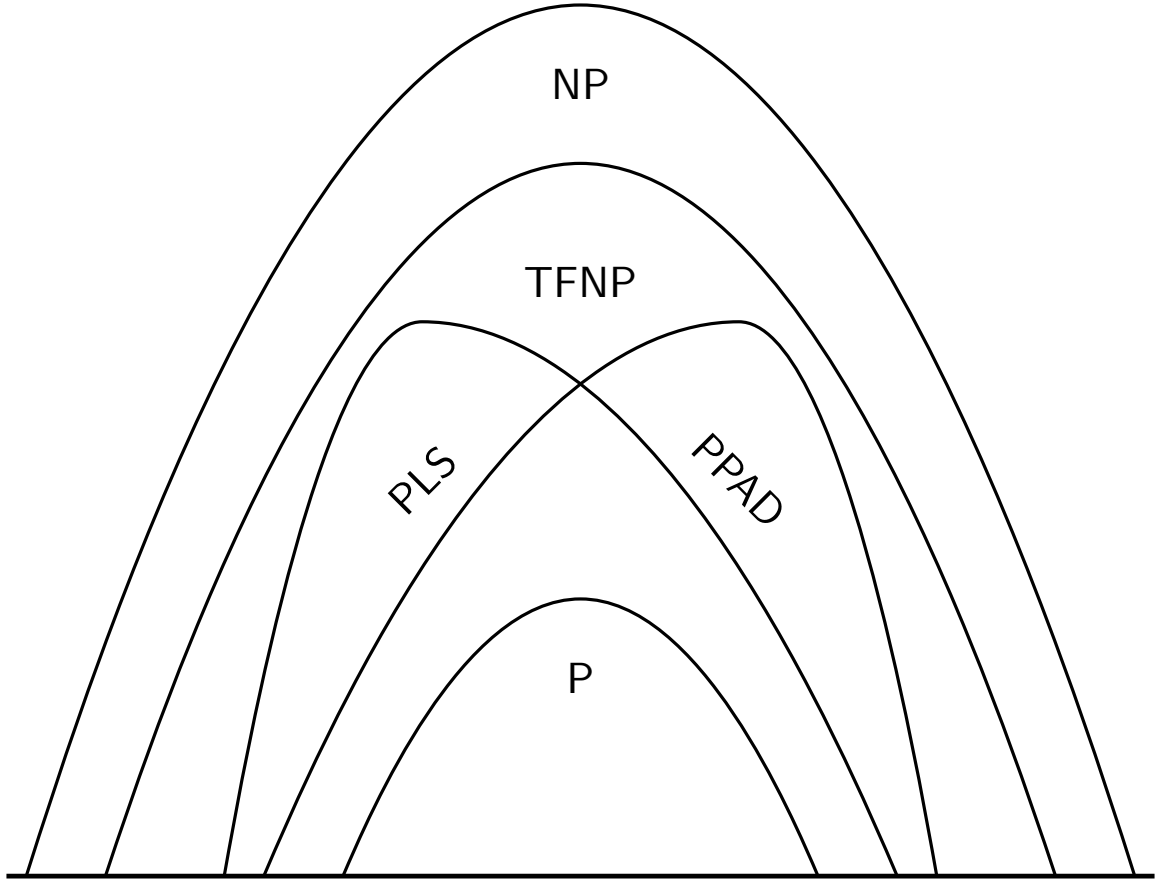
Simple stochastic games are in PLS



This puts simple-stochastic games **in PLS**

- A vertex of the DAG is a strategy for Max
- The neighbors are all strategies that can be switched to
- The potential is the sum of the values

Complexity classes between P and NP



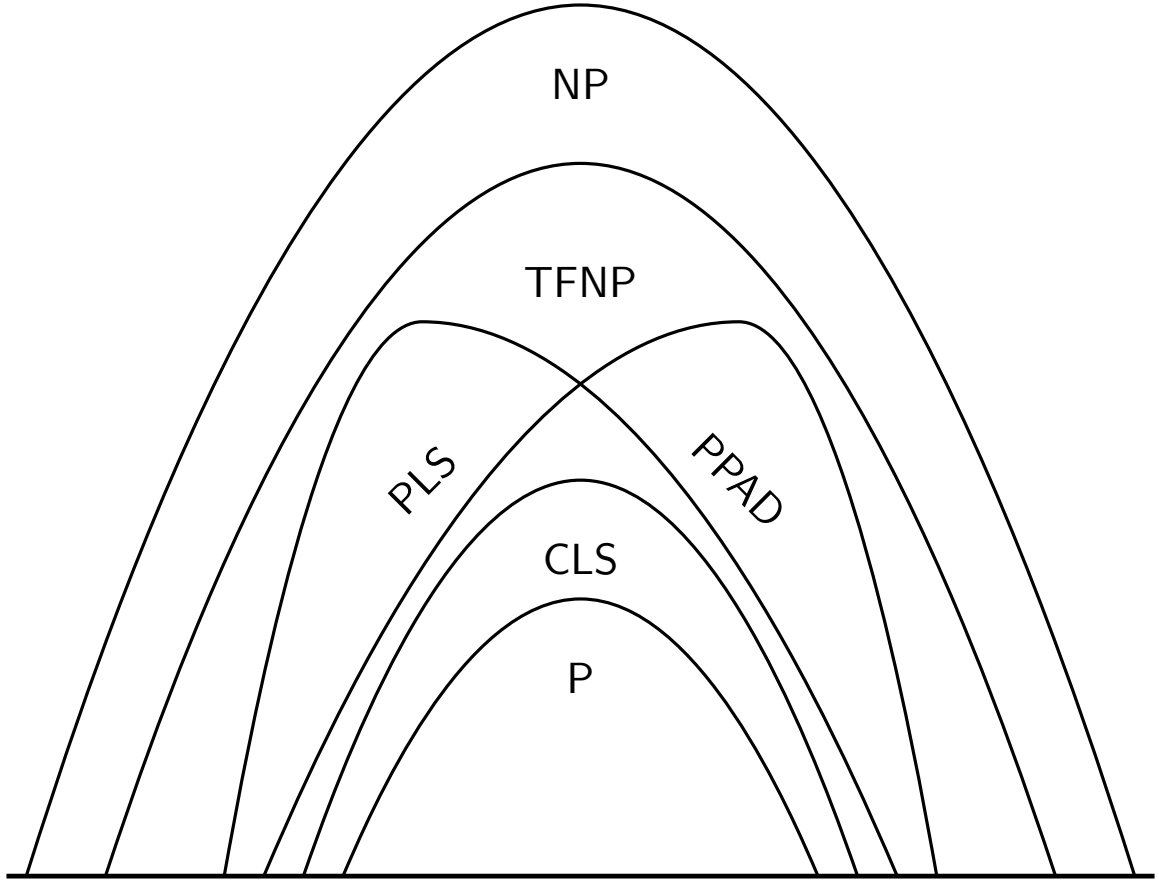
Are there any other interesting problems in PPAD and PLS?

Are there any other interesting problems in PPAD and PLS?

Yes!

1. Solving a Simple Stochastic Game
2. Finding a mixed NE of a Team Polymatrix Game
3. Finding a mixed NE of a Congestion Game
4. Solving a P-matrix Linear Complementarity Problem
5. Finding a fixed point of a Contraction Map
6. Solving reachability on a switching network
7. ...

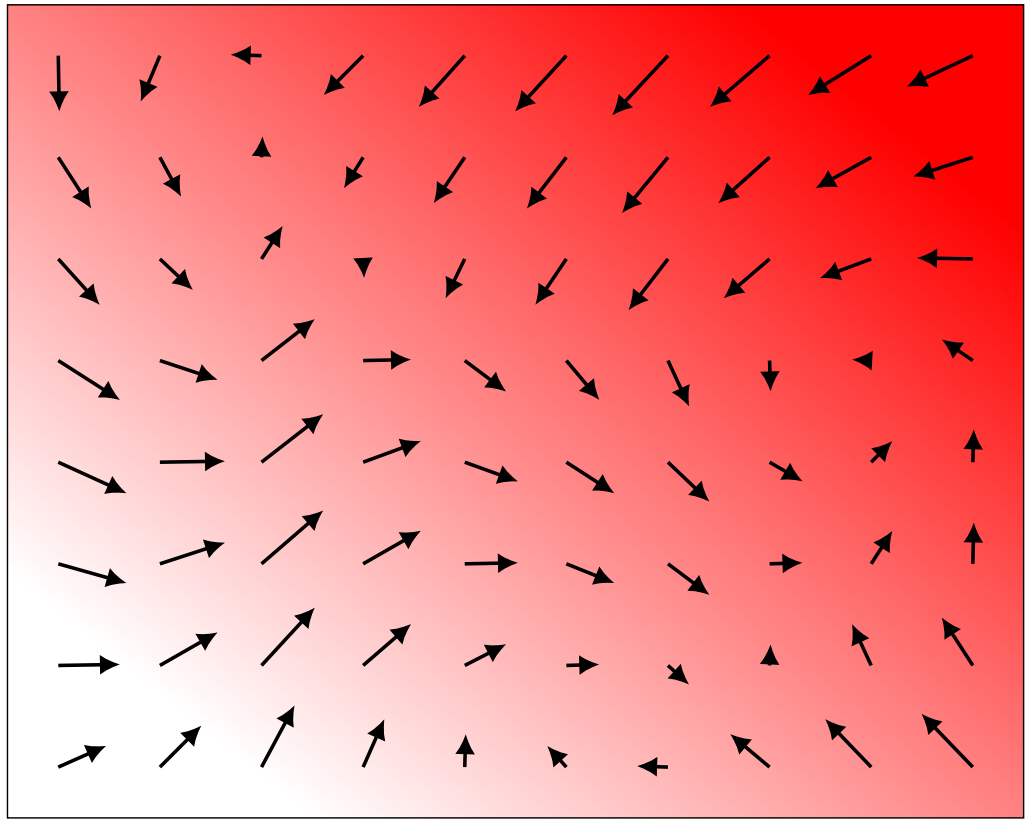
Complexity classes between P and NP



CLS was defined to capture these problems

(Daskalakis and Papadimitriou, 2011)

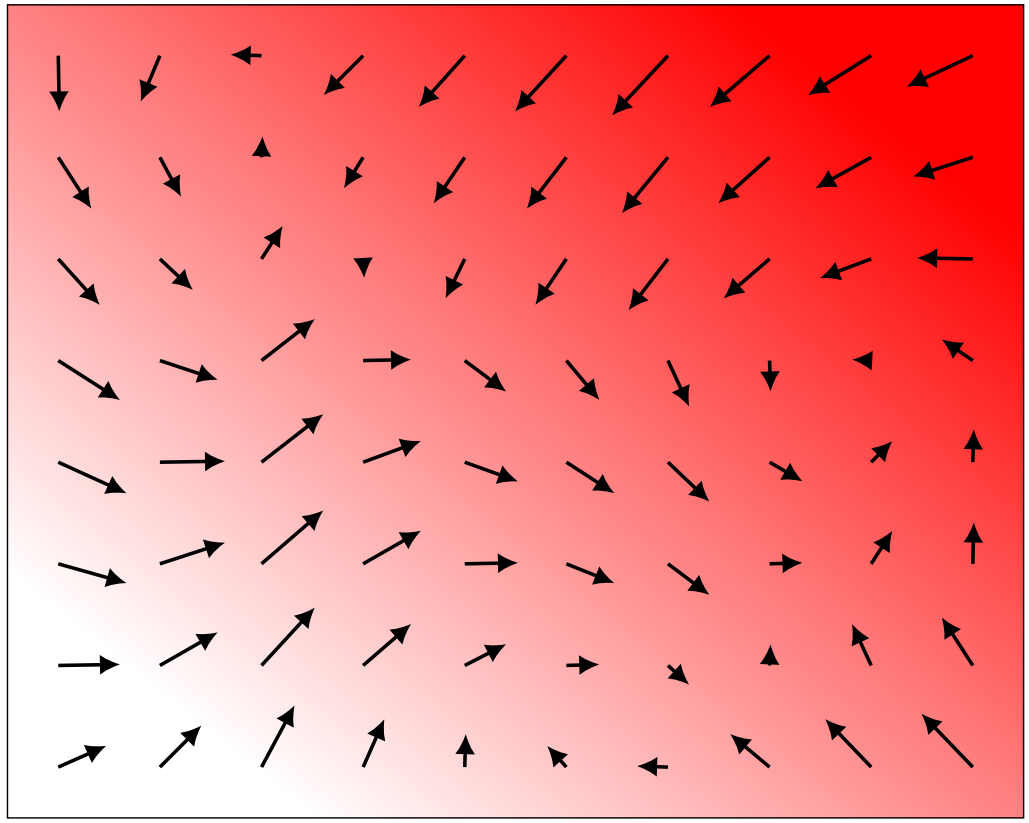
Continuous Local Search (CLS)



CLS is a Brouwer instance that **also** has a potential

- Continuous direction function $f : [0,1]^3 \to [0,1]^3$
- Continuous potential function $p : [0,1]^3 \to [0,1]$

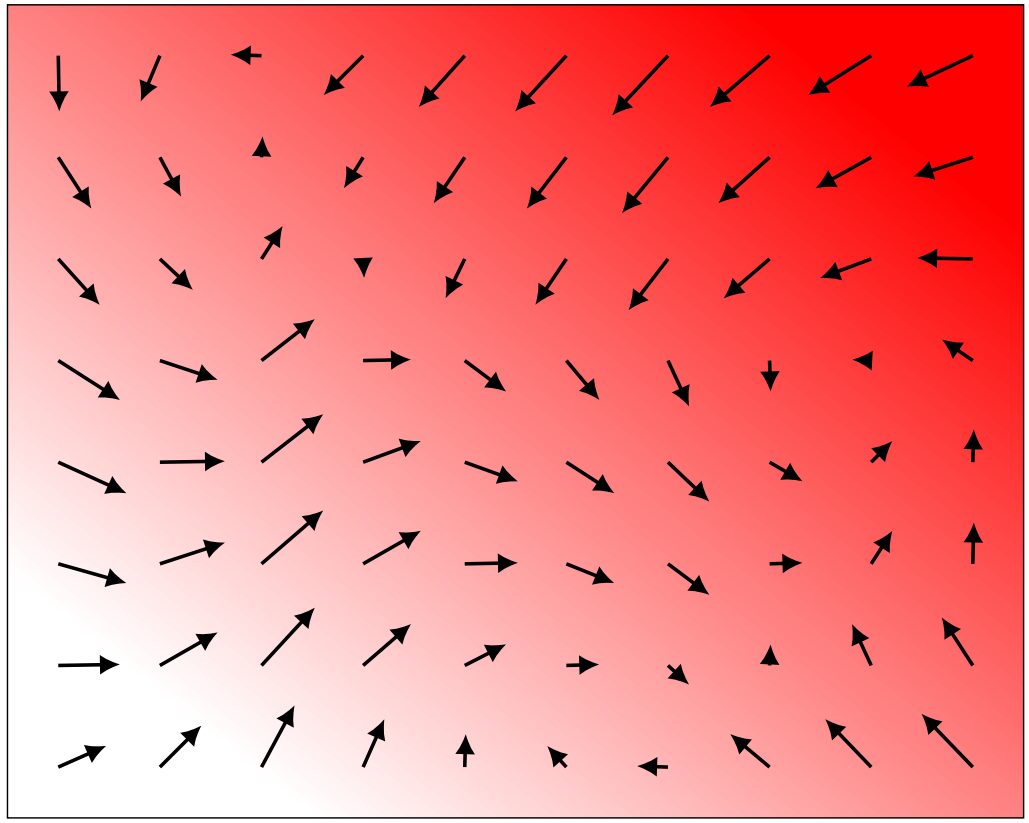
Continuous Local Search (CLS)



Find a point x where the potential **does not decrease**

$$p(f(x)) \geq p(x)$$

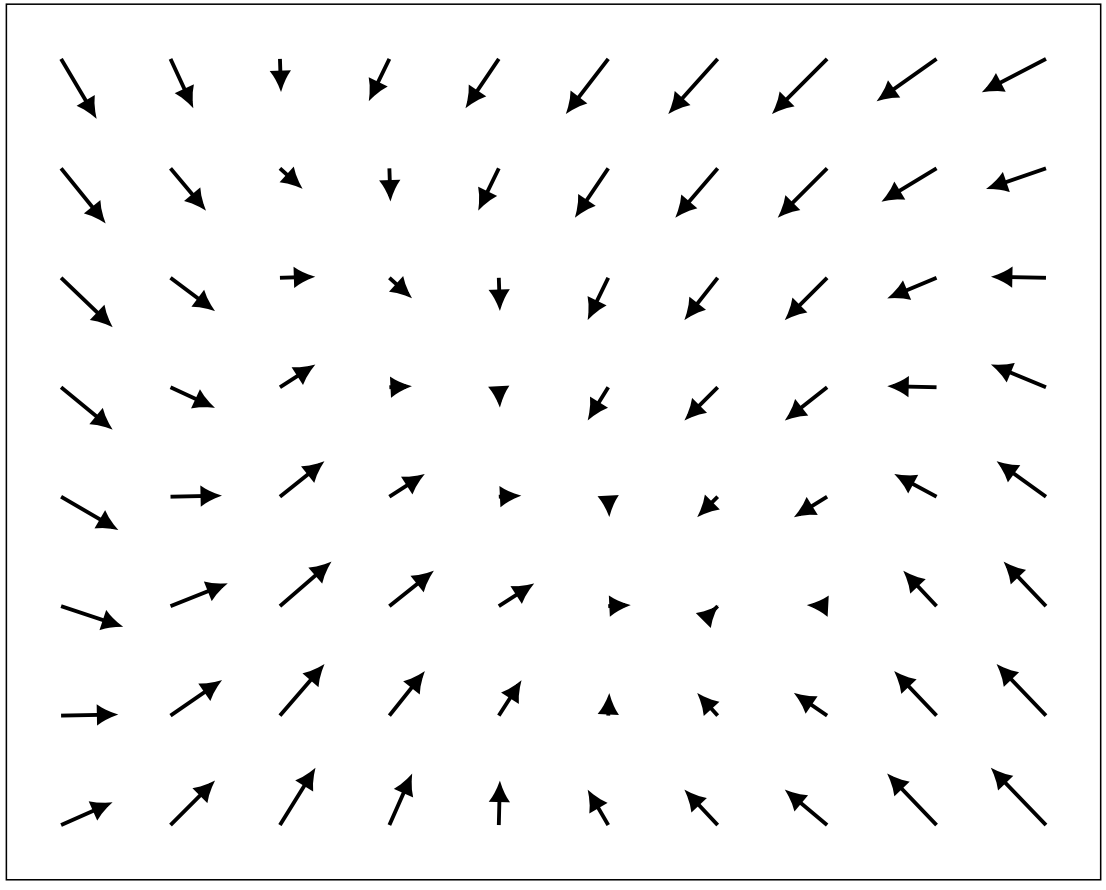
Continuous Local Search (CLS)



CLS contains **all** of the problems we saw on the previous slide

- It is now known to have **complete problems**
  (Daskalakis, Tzamos, Zampetakis, 2018) (FGMS, 2017)

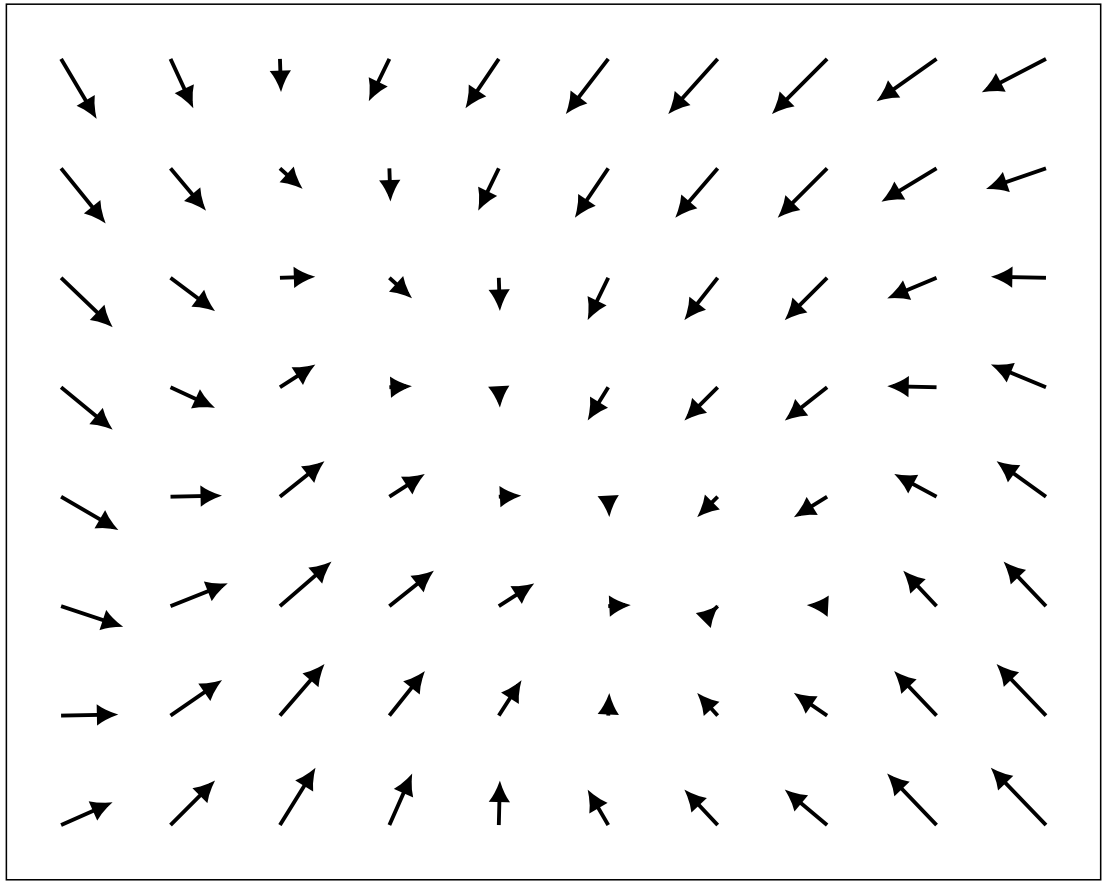
Contraction: a problem in CLS



f is **contracting** if

$$|f(x) - f(x')| \leq c \cdot |x - x'| \quad \text{for } c < 1$$

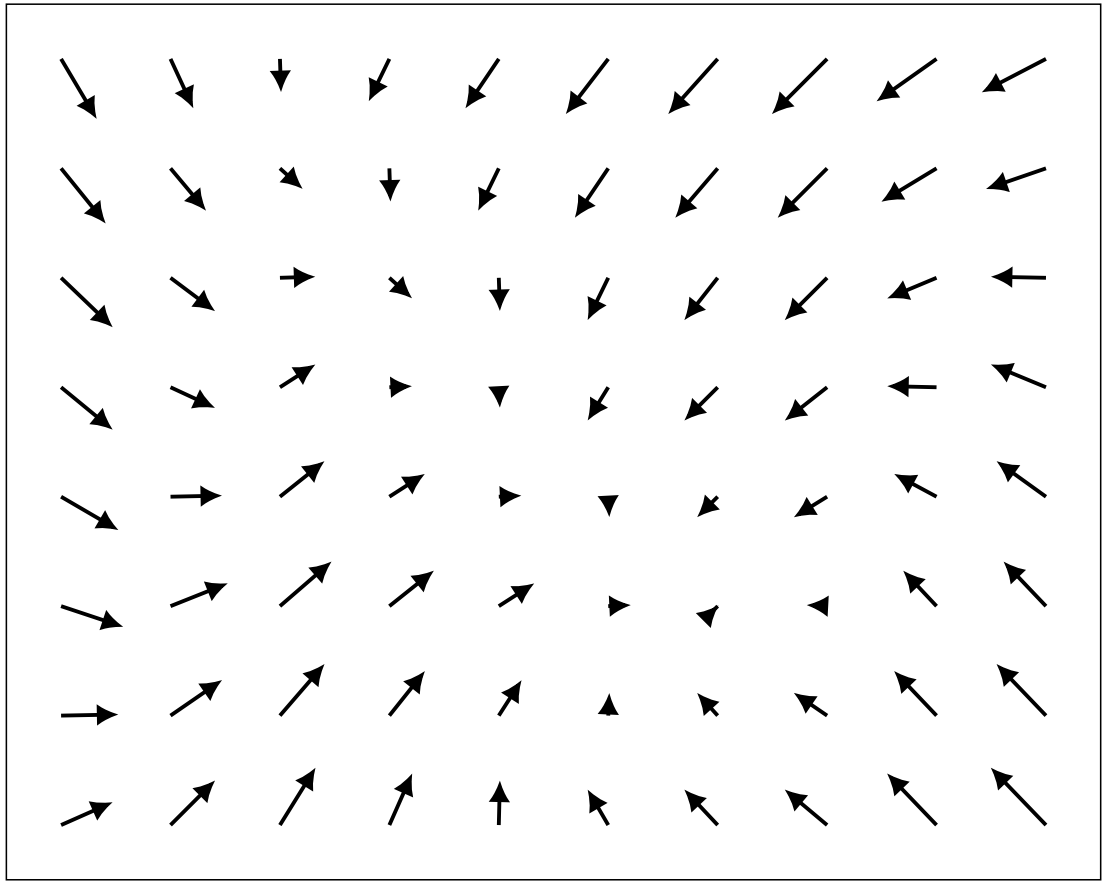
Contraction: a problem in CLS



Banach's fixpoint theorem

- Every contraction map has a **unique** fixpoint

Contraction: a problem in CLS

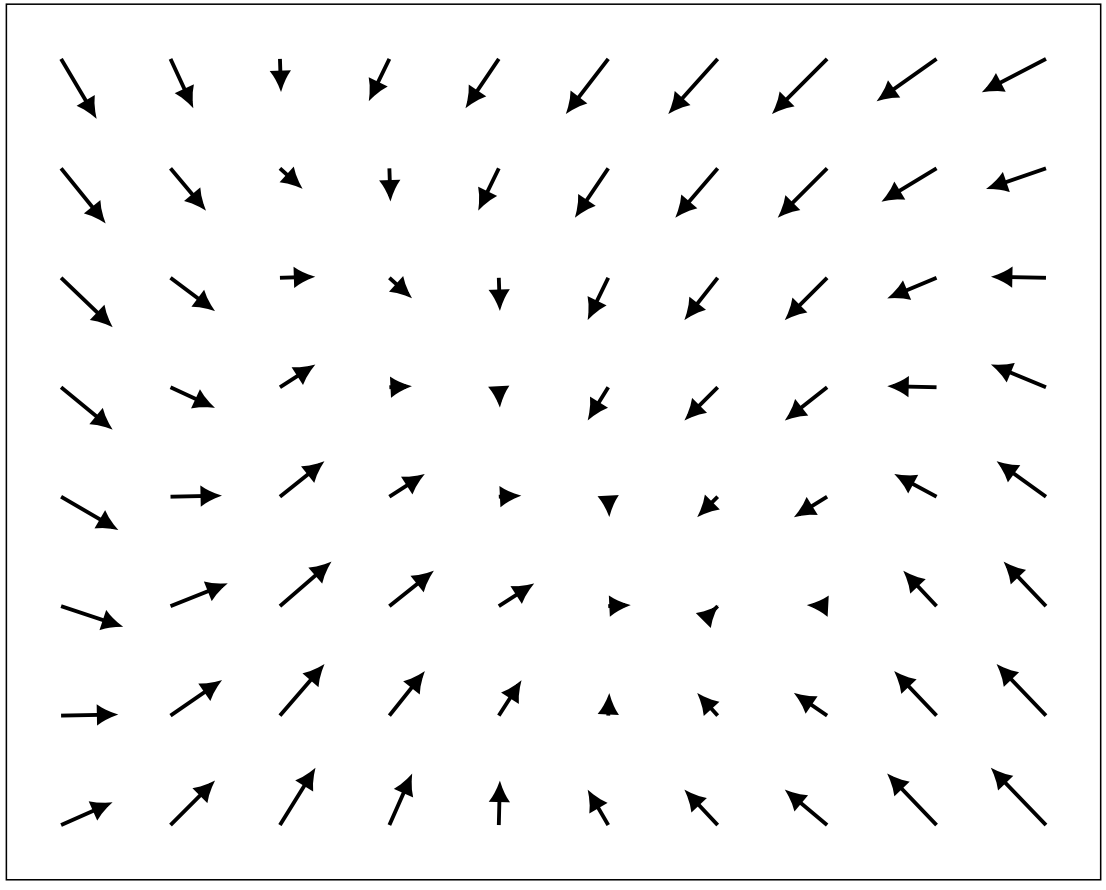


Problem: given a contraction map as an arithmetic circuit

- Find a **fixpoint** or a **violation** of contraction

No violations $\Rightarrow$ the problem has a **unique** solution

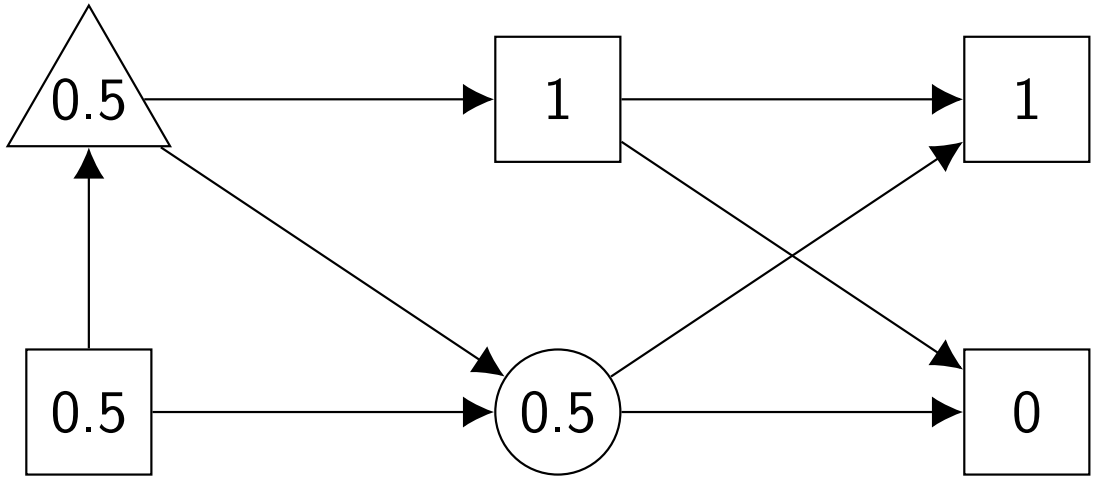
Contraction: a problem in CLS



Contraction is in CLS

- Direction function: f
- Potential function: $p(x) = |f(x) - x|$

Simple stochastic games are in CLS

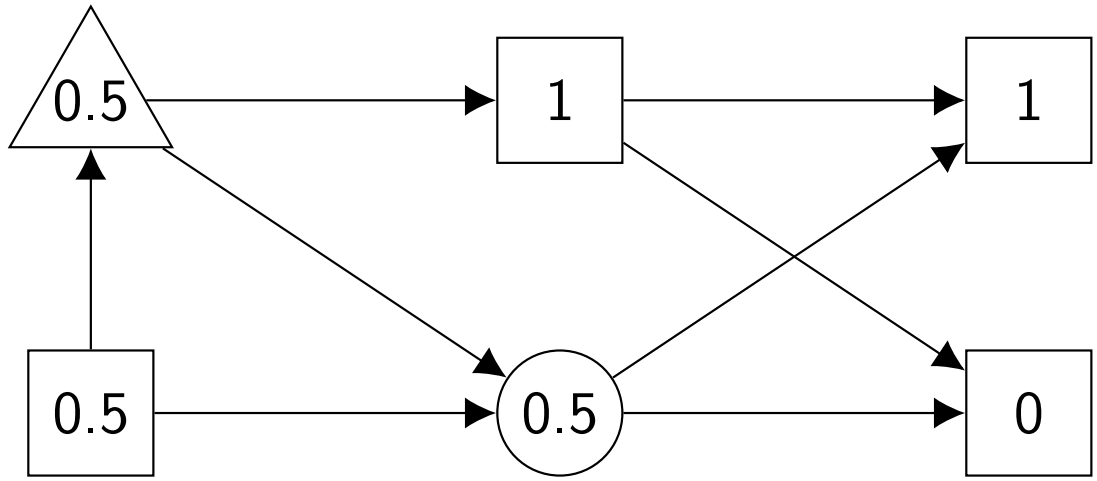


We can reduce SSG to contraction

- At each step **stop the game** with probability δ
- Otherwise continue

If δ is sufficiently small, the optimal values can be recovered

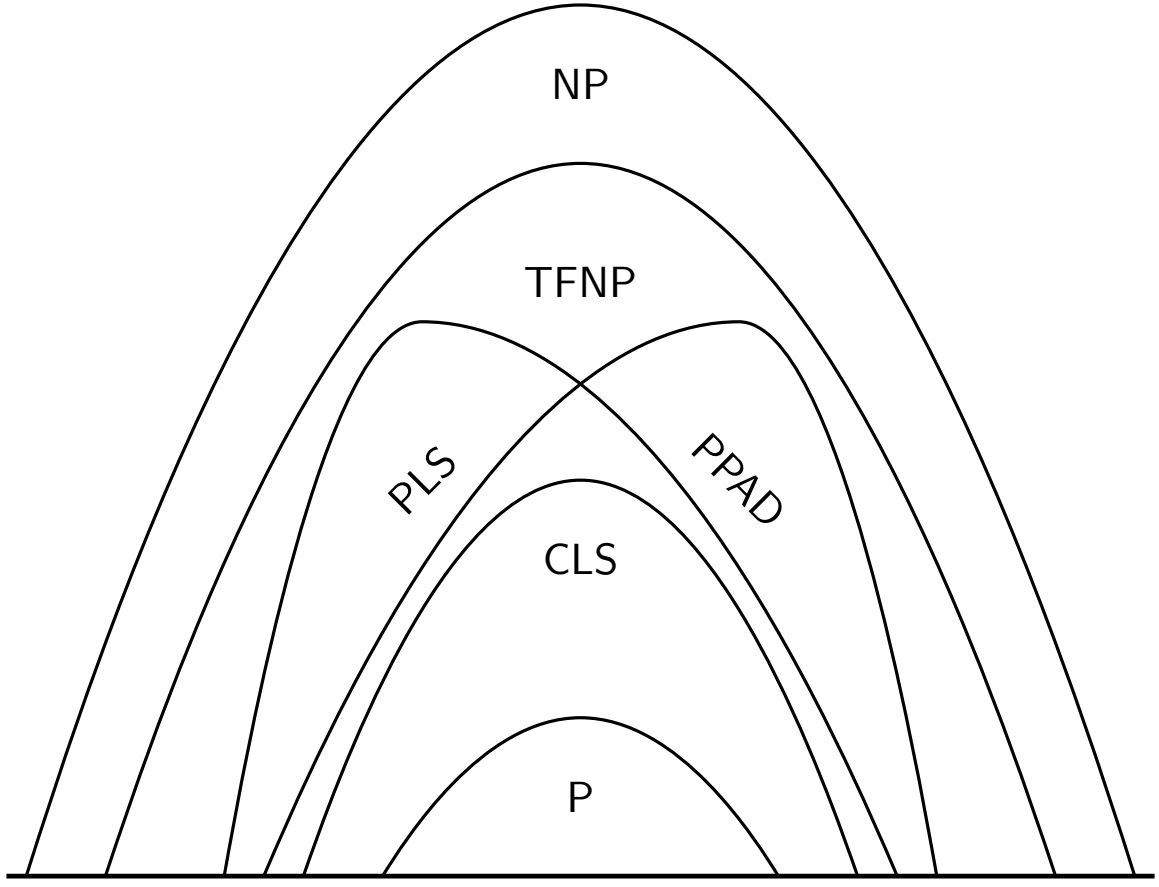
Simple stochastic games are in CLS



$$f(v) = \begin{cases} 1 & \text{if v is the target} \\ 0 & \text{if v is any other sink} \\ (1-\delta)\cdot\max(u,w) & \text{if v is a max vertex} \\ (1-\delta)\cdot\min(u,w) & \text{if v is a min vertex} \\ (1-\delta)\cdot(u+w)/2 & \text{if v is a random vertex} \end{cases}$$

f is contracting with $c = 1-\delta$

Complexity classes between P and NP



Are we done?

- Will all of our problems be CLS-complete?

Is CLS a good complexity class?

Positives

- Contains lots of problems
- Has complete problems

Negatives

- Currently has no **natural** complete problems

Natural problem:

did anyone care about it before you showed it was complete?

Unique End of Potential Line

Do we expect all of the problems in CLS to be complete?

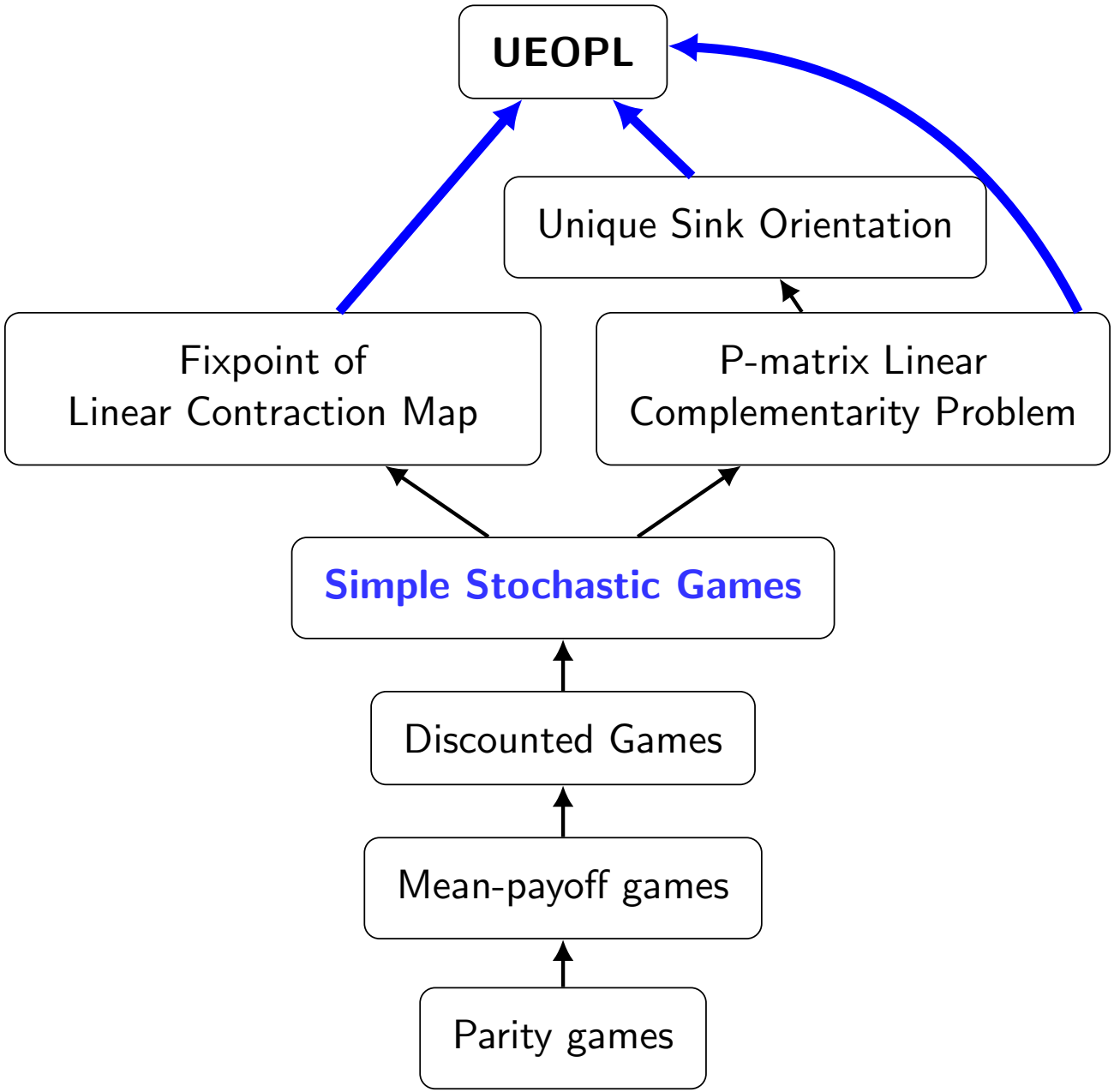
Our answer: **no!**

We identify a new class of problems **UEOPL** $\subseteq$ CLS containing

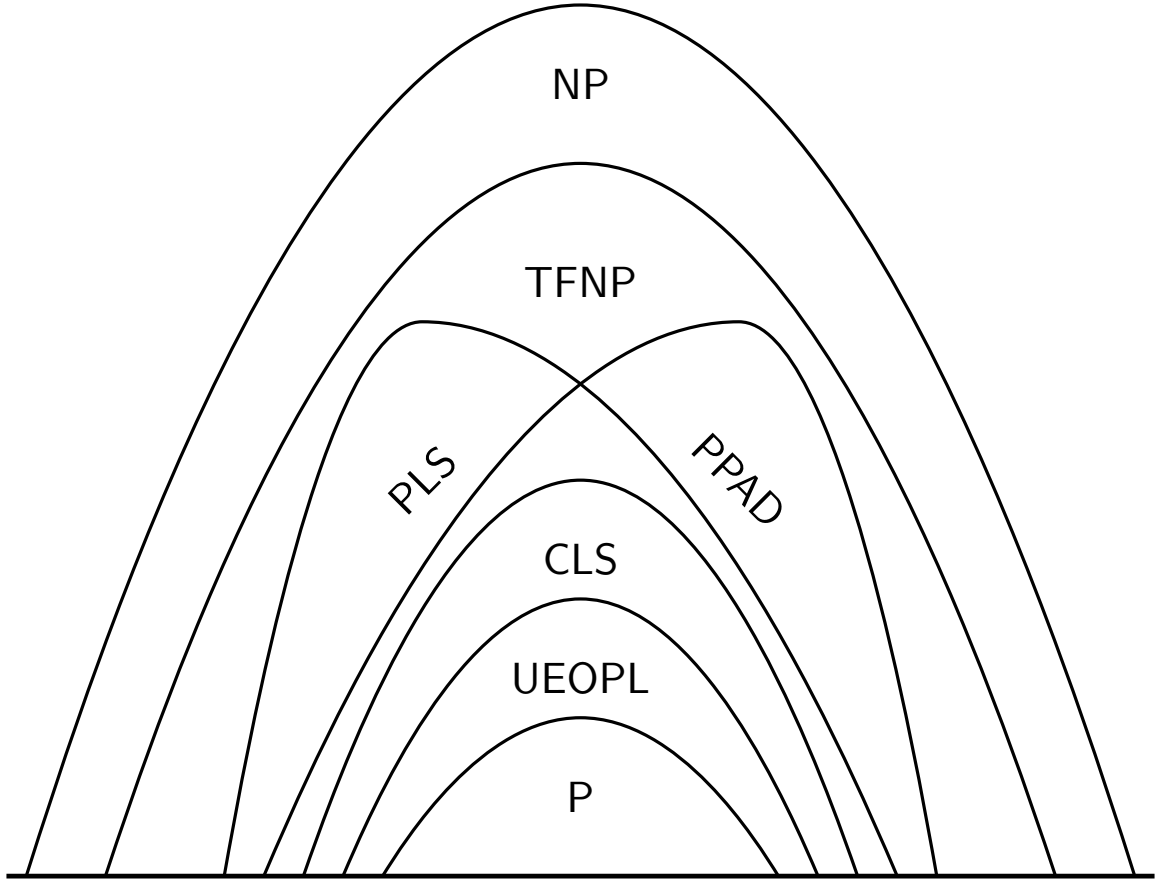
- Finding the fixpoint of a Linear Contraction Map
- Finding the sink of a Unique Sink Orientation
- Solving the P-matrix Linear Complementarity Problem

We believe that this is a **distinct** class of problems

Problems in UEOPL



Complexity classes between P and NP



UEOPL is the closest class to P among the subclasses of TFNP

Our **three problems**

- Contraction
- Unique sink orientation
- P-matrix LCP

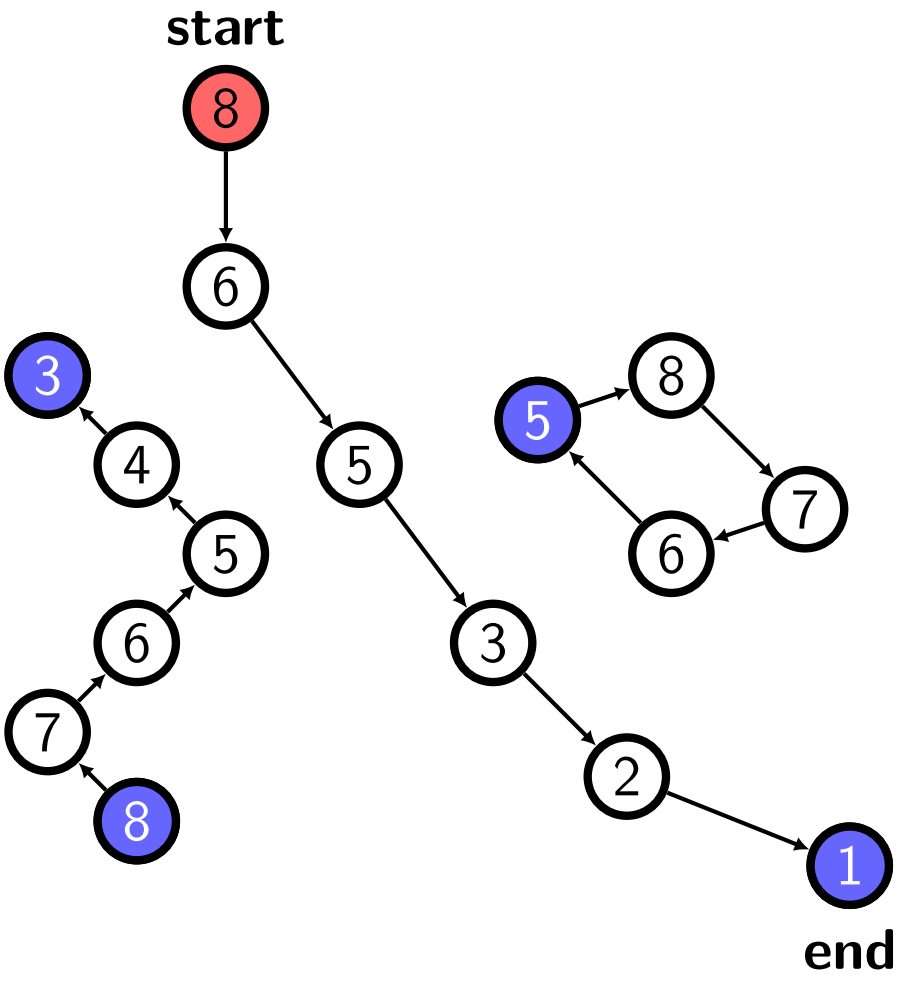
Each can be formulated so that there are

- **proper solutions**
- **violation solutions**

When there are no **violations** there is a **unique** solution

UEOPL is intended to capture problems like this

End of Potential Line (EOPL)



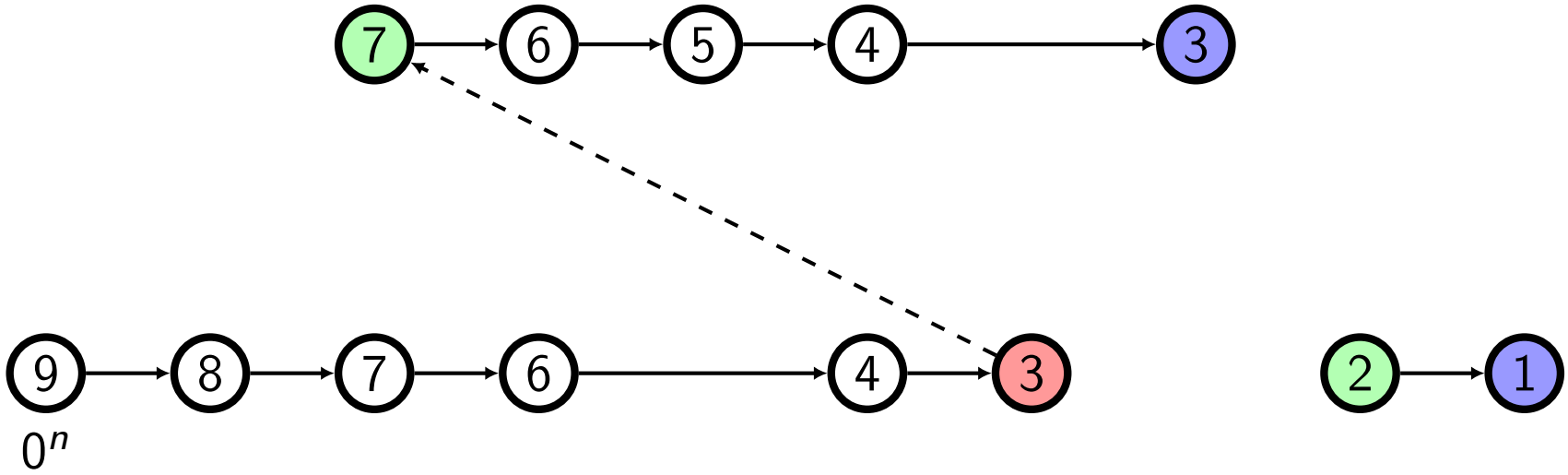
Combines the two **canonical** complete problems

- An End-of-the-Line instance
- That has a potential

Find

- The end of a line
- A vertex where the potential increases

Unique End of Potential Line (UEOPL)

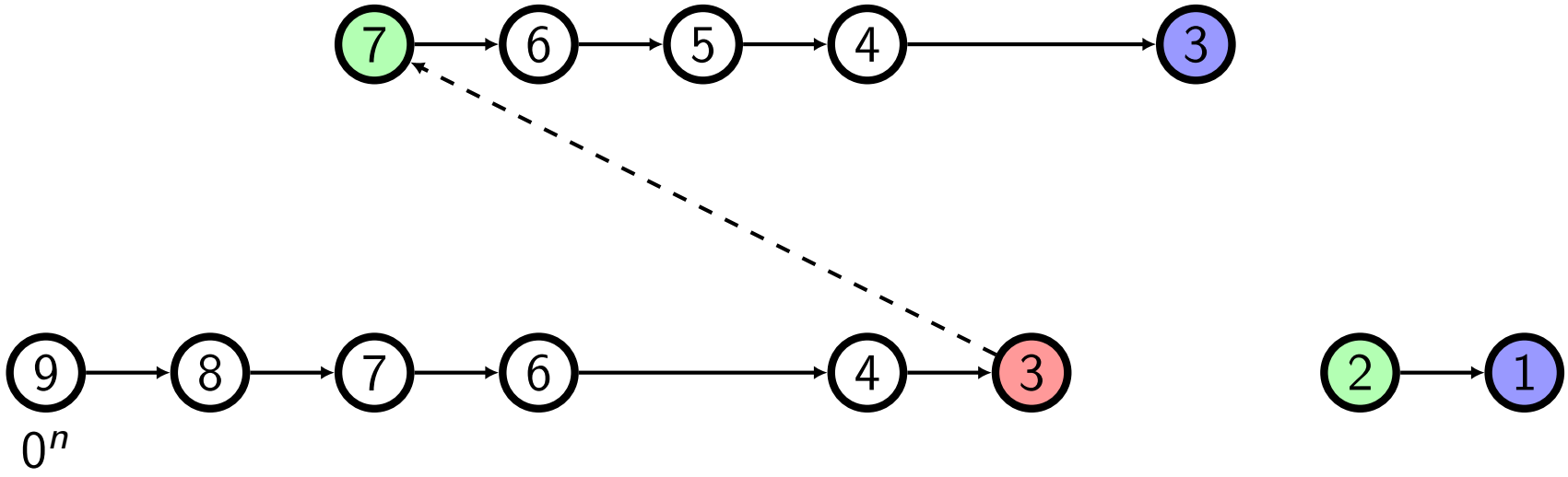


An EOPL instance with violations to make sure the line is unique

- Proper solution: The end of a line
- Violation 1: The start of a line other than 0^n
- Violation 2: An edge that increases the potential
- Violation 3: Any pair of vertices v and u satisfying

$$P(v) \geq P(u) > P(S(v))$$

Unique End of Potential Line (UEOPL)



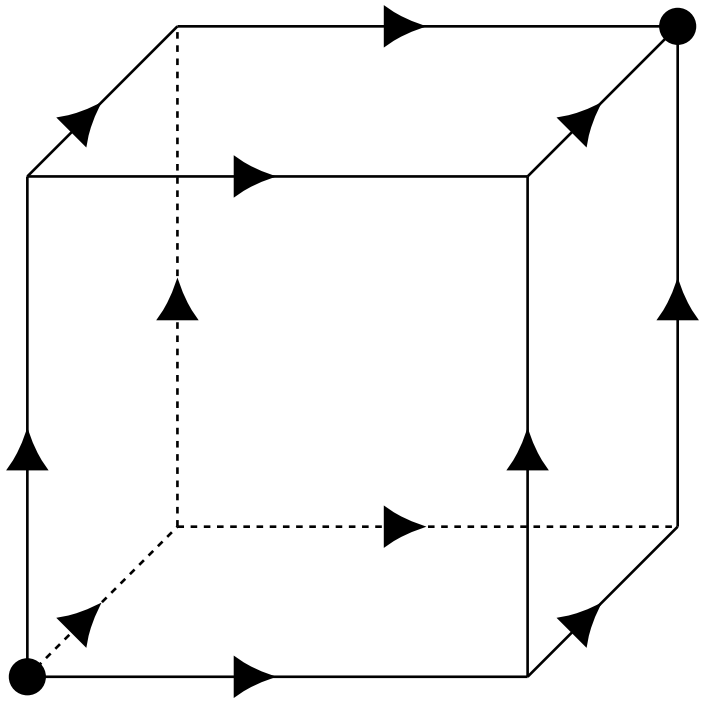
If there are no violations then there is a **unique** line

- That starts at at 0^n
- And ends at the **unique** proper solution to the problem

Unique Sink Orientations of Cubes

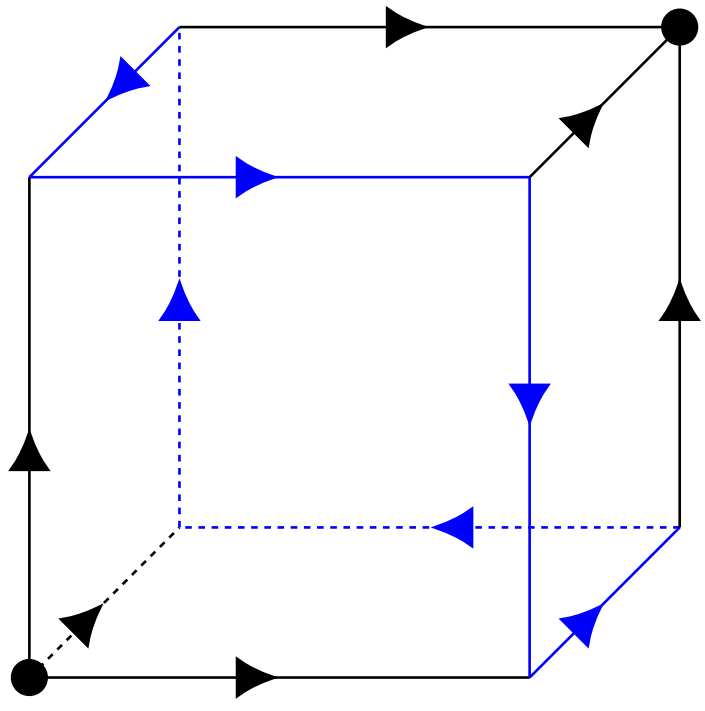
Orient the edges of an n-dimensional cube

- So that every face has a **unique** sink

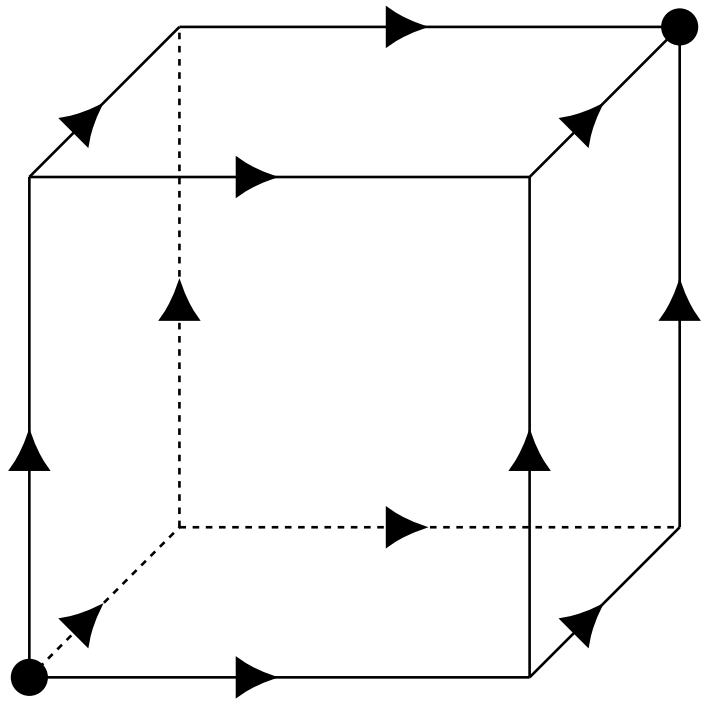


Unique Sink Orientations of Cubes

Can be **cyclic**:



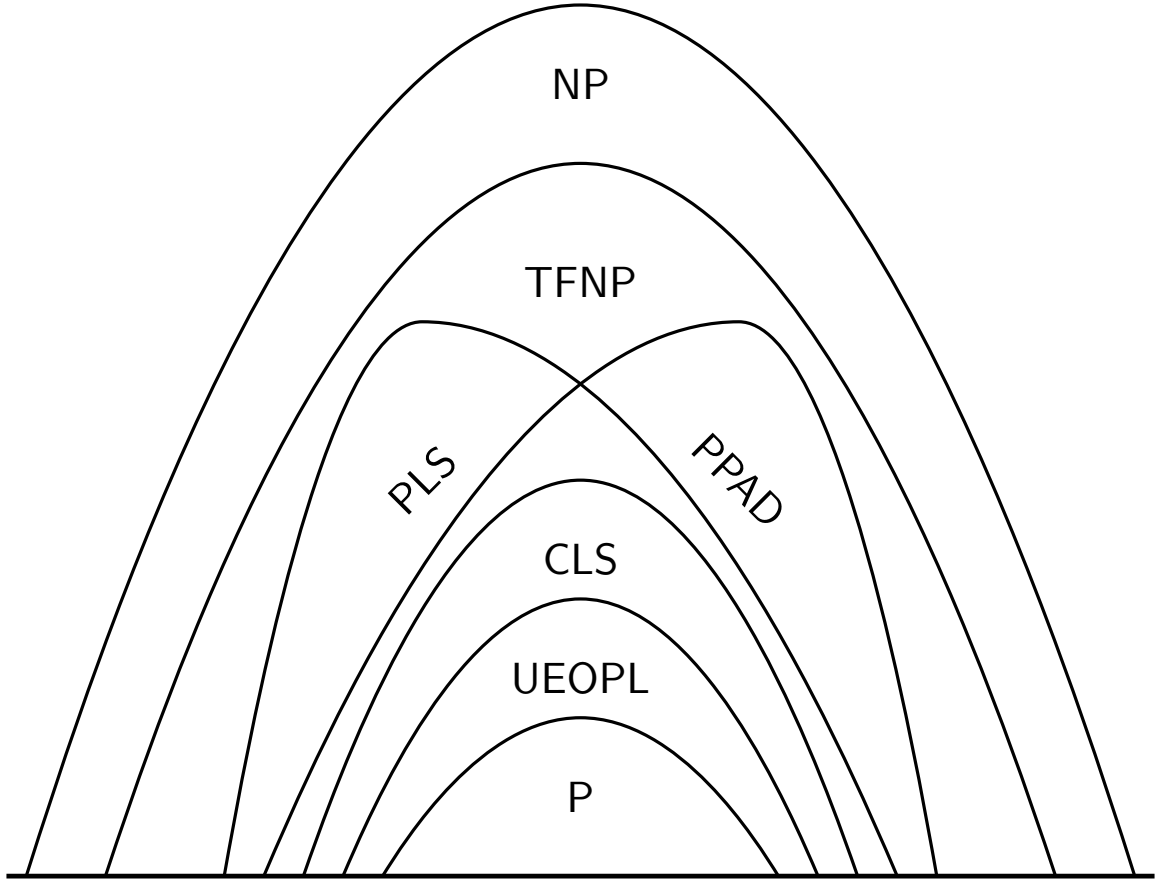
Unique Sink Orientations of Cubes



Simple-stochastic games reduce to **acyclic** USO

- The strategy improvement algorithm gives an AUSO
- Vertices are strategies
- The orientation is determined by the potential

Complexity classes between P and NP



Conjectures

- USO is complete for UEOPL
- Contraction is complete for UEOPL
- PLCP is complete for UEOPL

Most of our algorithms for SSGs actually solve one of these problems

Simple-stochastic games are **not** complete for UEOPL

SSGs are not a promise problem!

Open questions

What is the complexity of solving a simple stochastic game?

Is it **hard?**

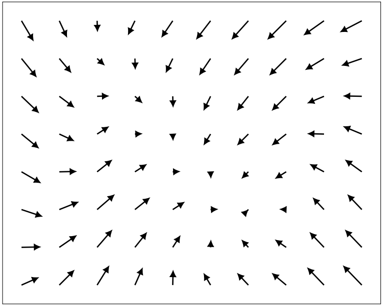
- Then what is the right class?

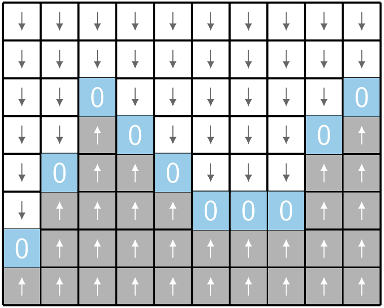
Is it **easy?**

- Then show me the algorithm!

Thanks!

Contraction to UEOPL

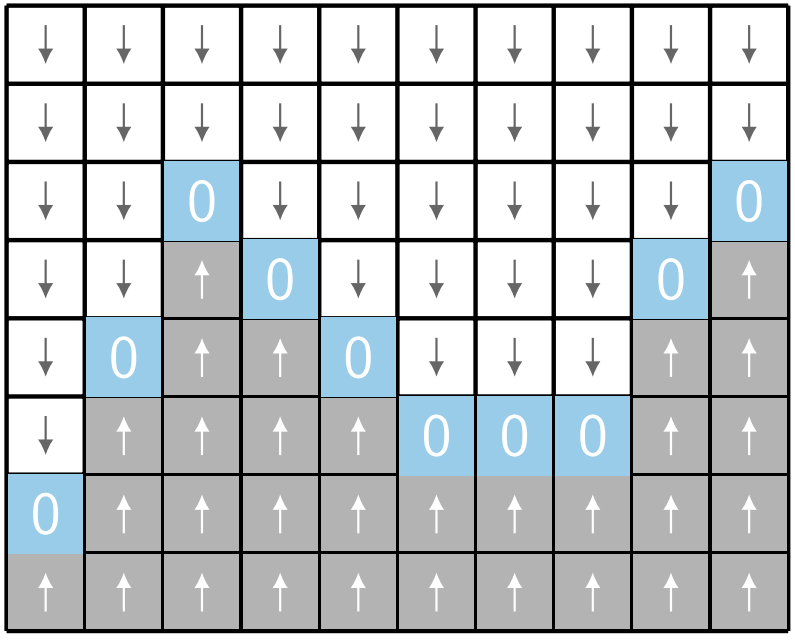


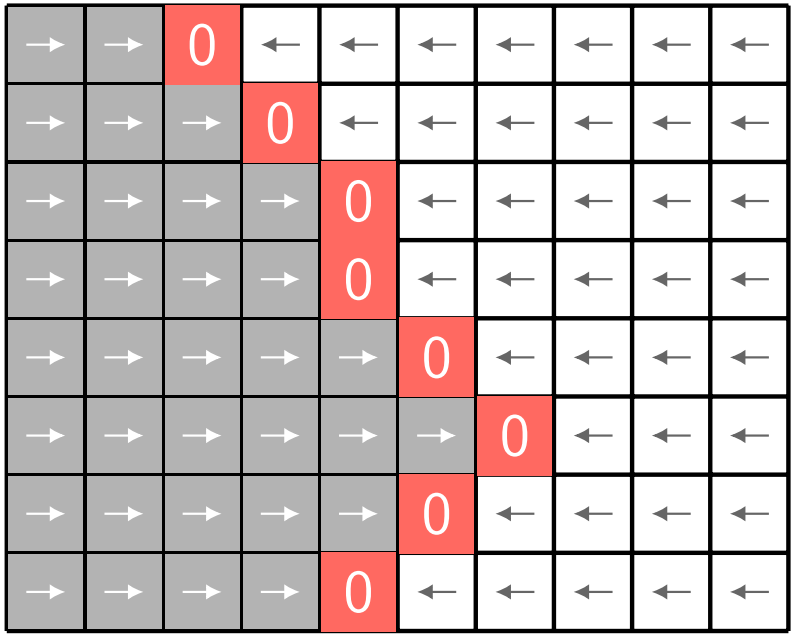


First we discretize the problem

- Lay a grid of points over the space
- For each dimension construct a **direction** function

Contraction to UEOPL

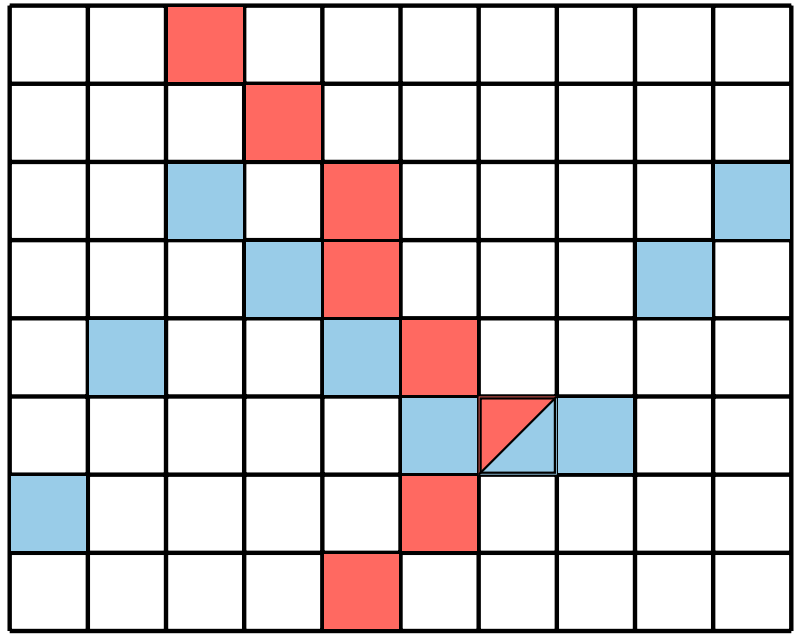




Discrete contraction

- Find a point that is 0 in all dimensions

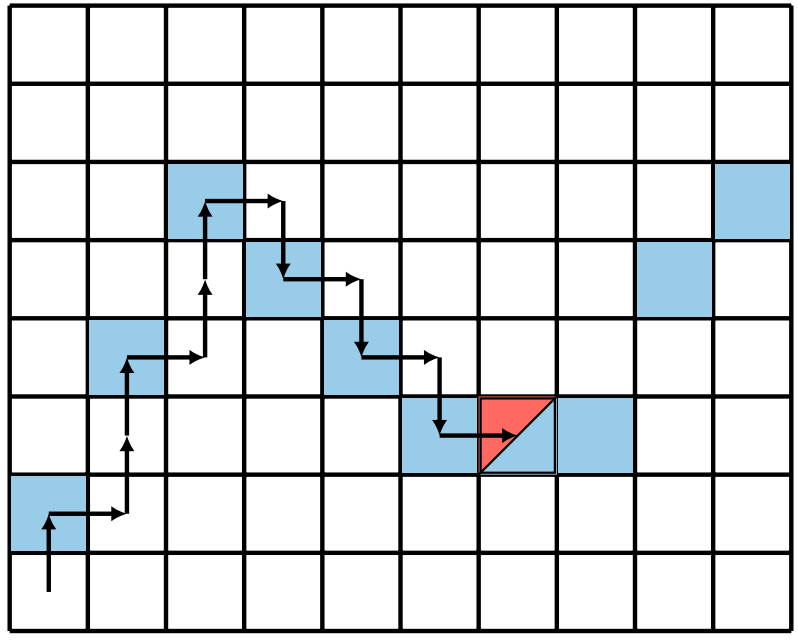
Contraction to UEOPL



A point is on the **surface** if it is 0 for some direction

- Every left/right slice has a unique point on the blue surface
- At each of these, we can follow the red direction function

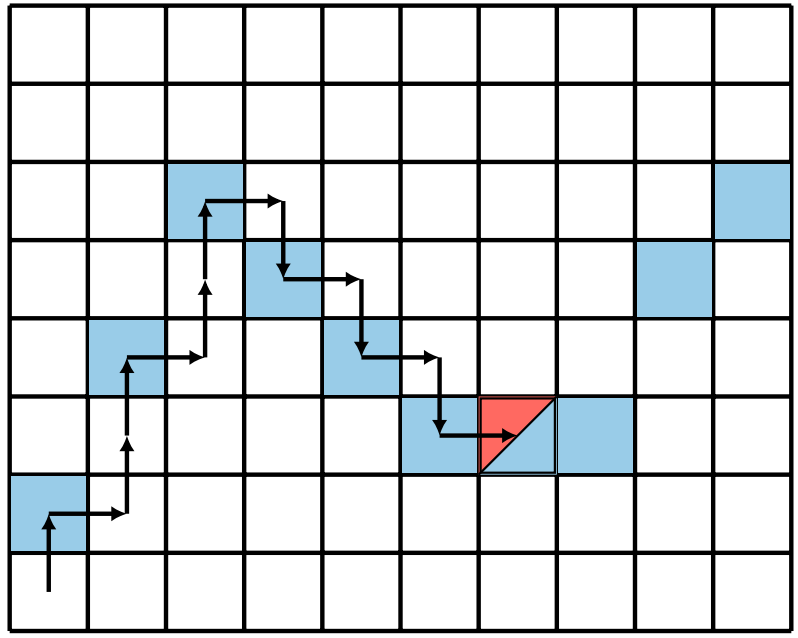
Contraction to UEOPL



The path

1. Start at $(0, 0)$
2. Find the blue surface
3. Take one step in the red direction
4. If not at red surface, go to 2

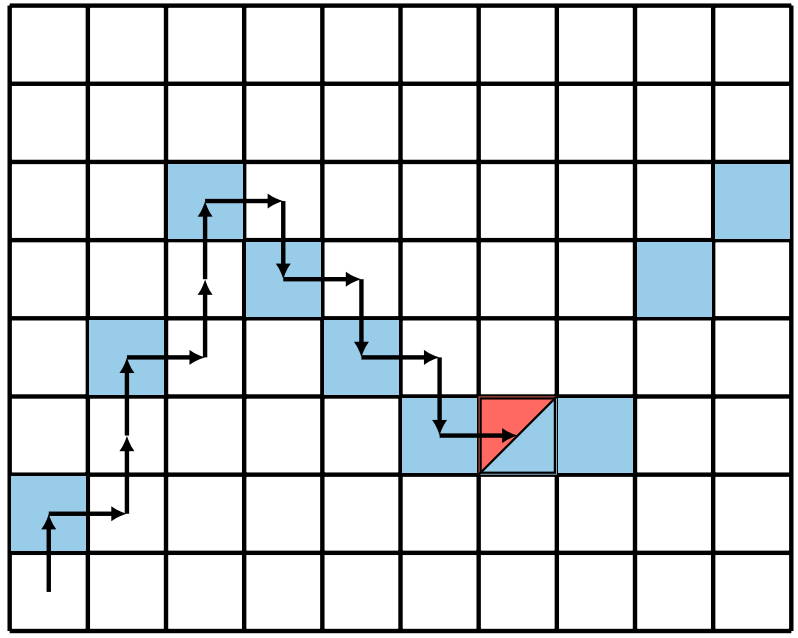
Contraction to UEOPL



The potential

- The path never moves left
- In every slice, it either moves moves up or down

Contraction to UEOPL

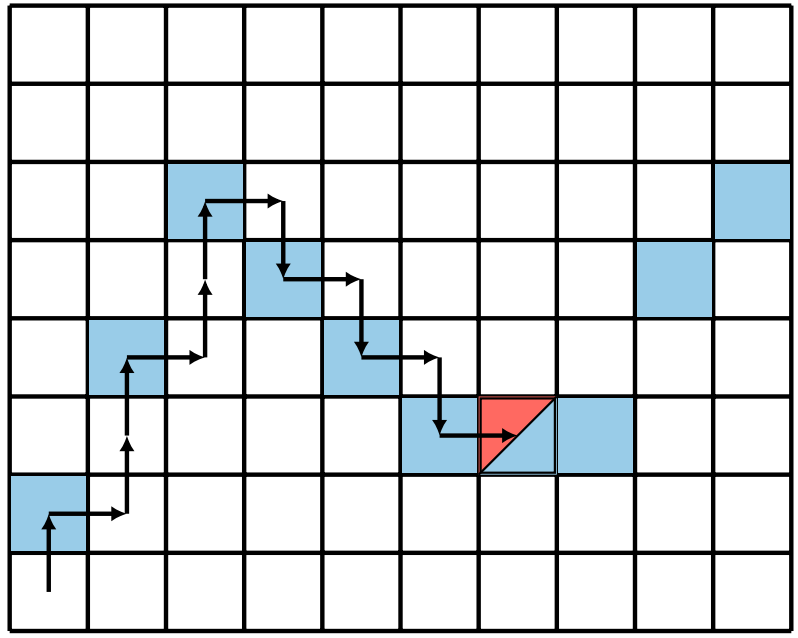


So we can use a pair (a, b) ordered lexicographically where

- a is the x coordinate of the vertex
- b is
  - y if we are moving up
  - $-y$ if we are moving down

This monotonically increases along the line

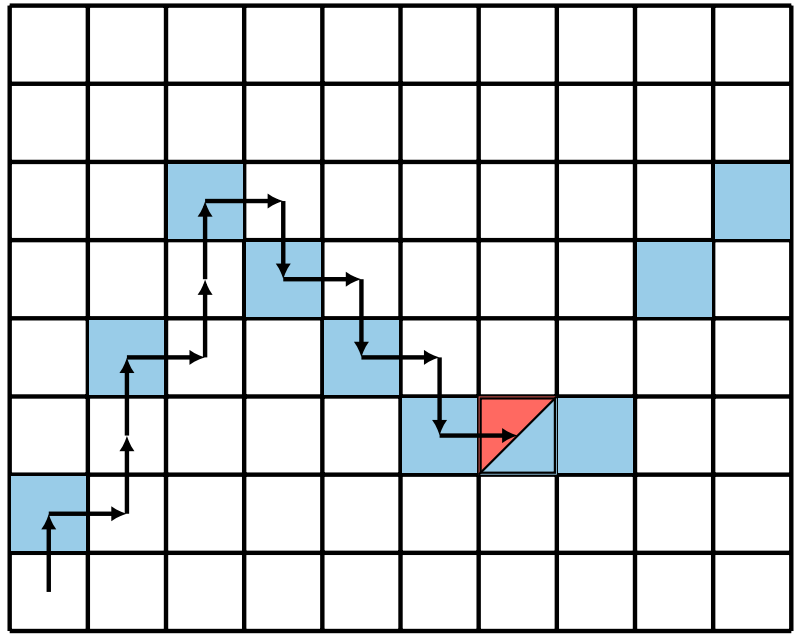
Contraction to UEOPL



Actually, this formulation only gives us a **forward** circuit

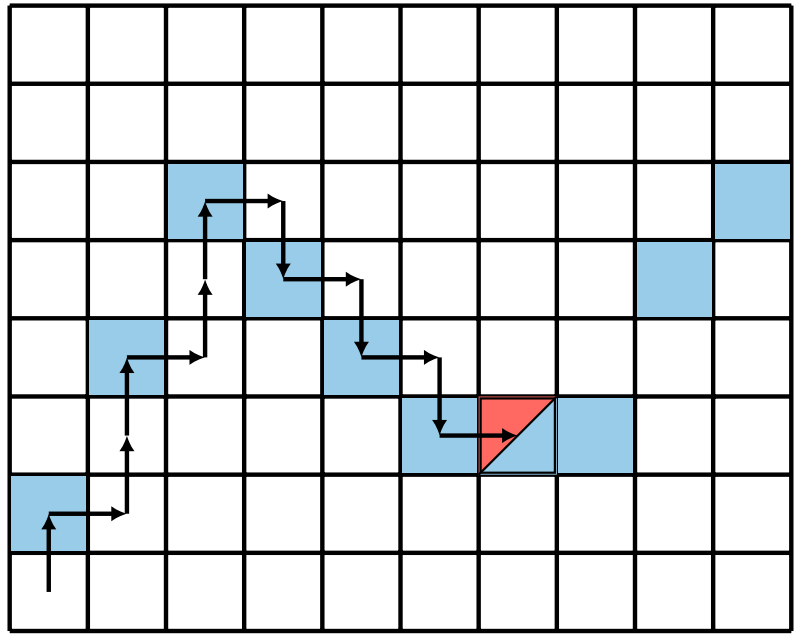
- But the line is unique
- So we can apply a technique of Hubáček and Yogev (2017) to make the line reversible

Contraction to UEOPL



This generalises to arbitrary dimension

Contraction to UEOPL



Theorem

Contraction is in UEOPL