END OF POTENTIAL LINE

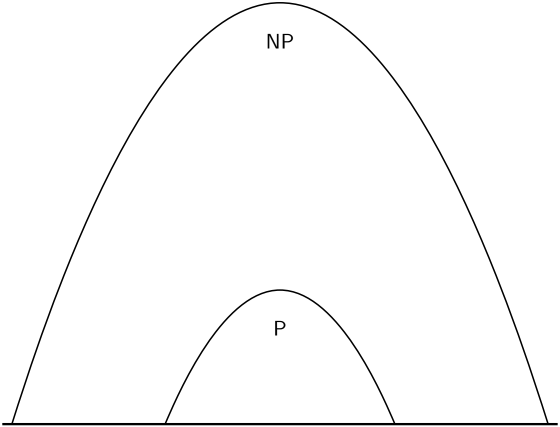
John Fearnley[1] Spencer Gordon[2] Ruta Mehta[3]
Rahul Savani[1]

[1]University of Liverpool

[2]Caltech

[3]University of Illinois at Urbana-Champaign

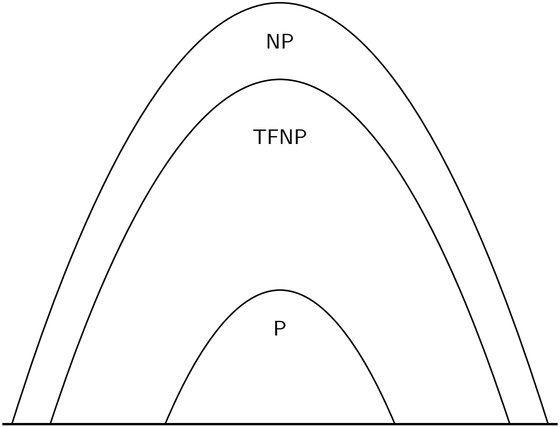
Complexity classes between P and NP



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

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

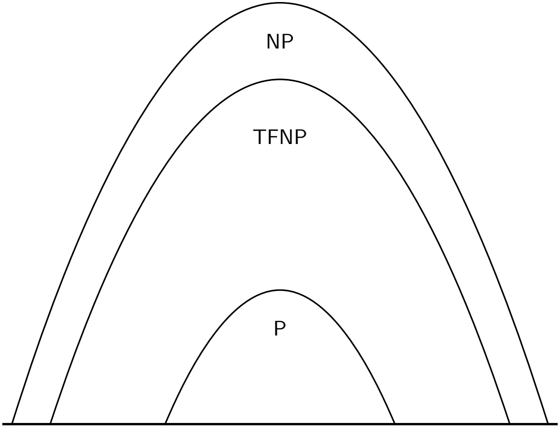
Complexity classes between P and NP



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

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

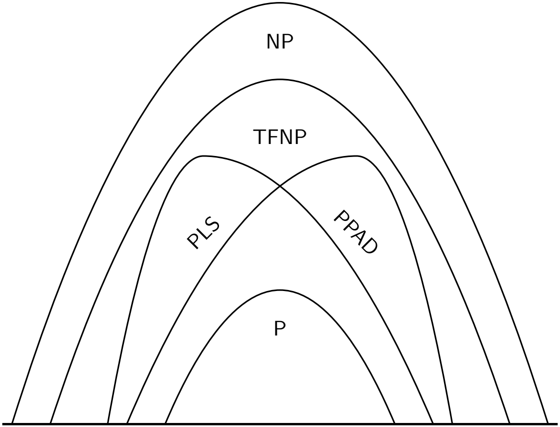
Complexity classes between P and NP



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

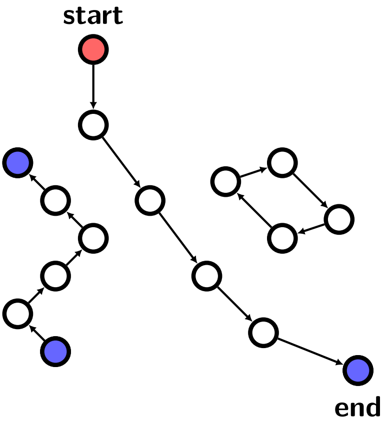
- Contains factoring, Nash equilibria, local max cut, simple-stochastic games, ...

Complexity classes between P and NP



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

PPAD (Papadimitriou 1994)

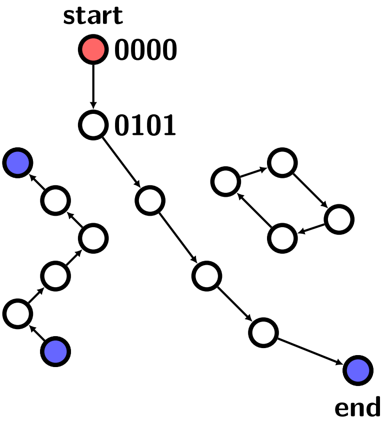


End-of-the-Line:

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**

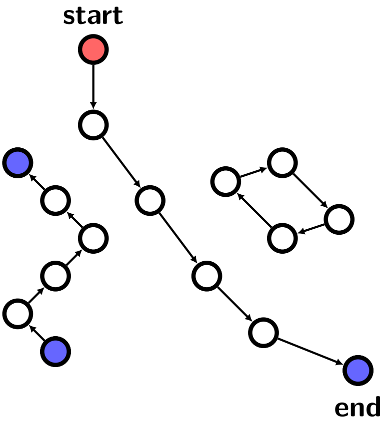
It is defined by

- A circuit $\boldsymbol{S}$ that gives a successor
- A circuit $\boldsymbol{P}$ that gives a predecessor

$$\boldsymbol{S}(0000) = 0101$$

$$\boldsymbol{P}(0101) = 0000$$

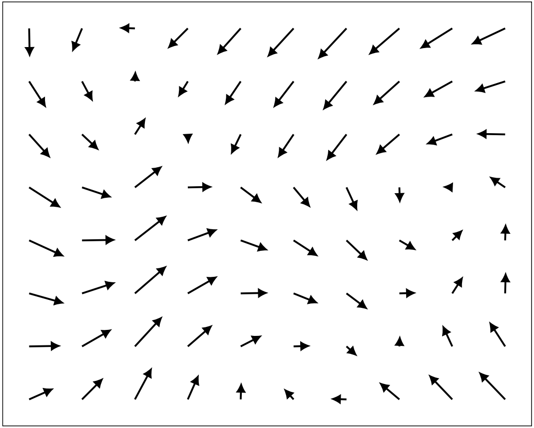
PPAD (Papadimitriou 1994)



Problem ***A*** is

- in PPAD if ***A*** reduces to EOTL
- PPAD-complete if EOTL also reduces to it

Brouwer: A PPAD-complete problem



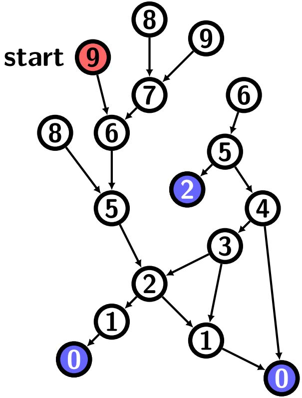
Given a continuous function $f : [0,1]^2 \to [0,1]^2$

- 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

Polynomial Local Search (PLS)



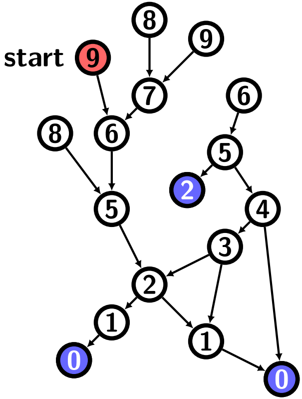
Given

- a DAG
- a starting vertex

Find

- a sink vertex

Polynomial Local Search (PLS)



Catch:
The graph is **exponentially large**

Defined by

- A circuit $\boldsymbol{S}$ giving the successor vertices
- A circuit $\boldsymbol{p}$ giving a **potential**

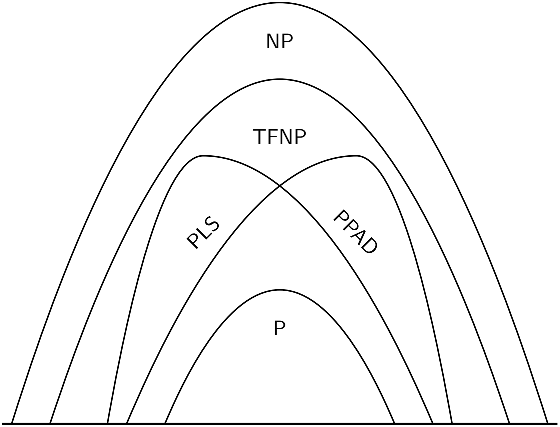
Every edge decreases the potential

$$\boldsymbol{p}(\boldsymbol{S}(\boldsymbol{v})) < \boldsymbol{p}(\boldsymbol{v})$$

PLS-complete problems:

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

Complexity classes between P and NP



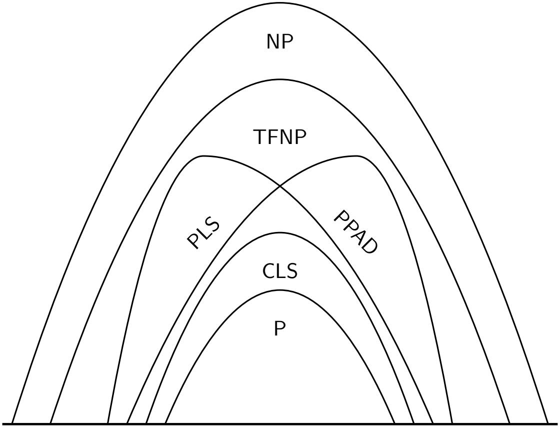
Are there interesting problems in PPAD and PLS?

Are there interesting problems in PPAD and PLS?

Yes!

1. Finding a mixed NE of a Team Polymatrix Game
2. Finding a mixed NE of a Congestion Game
3. Solving a Simple Stochastic 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

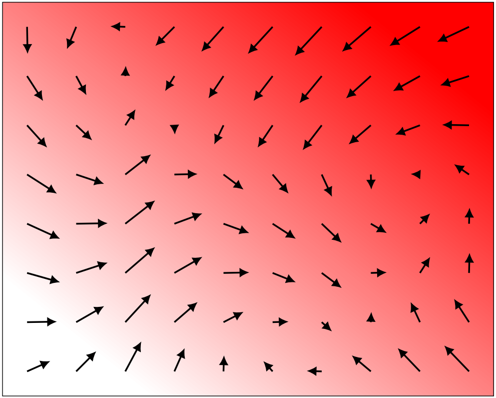
Complexity classes between P and NP



CLS was defined to capture these problems

(Daskalakis and Papadimitriou, 2011)

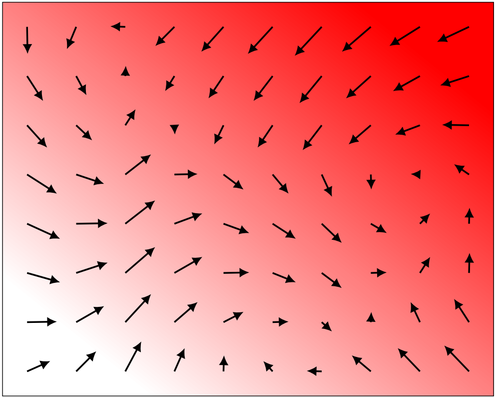
Continuous Local Search (CLS)



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

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

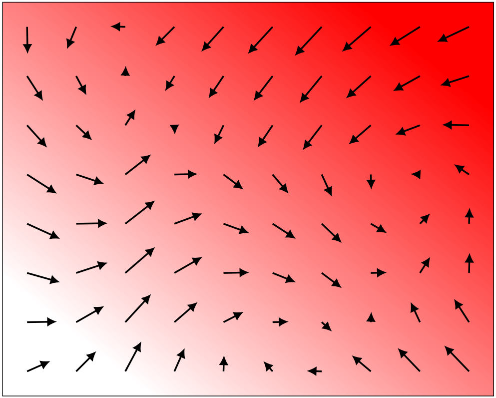
Continuous Local Search (CLS)



Find a point $\boldsymbol{x}$ 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

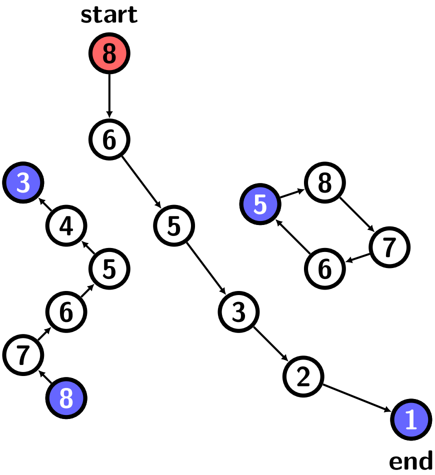
CLS combines

- the **continuous** PPAD-complete problem Brouwer
- the **canonical** PLS-complete problem

This work

Why not combine both canonical problems?

End of Potential Line (EOPL)



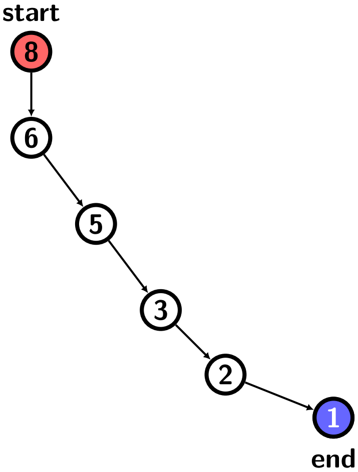
Combines the two **canonical** complete problems

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

Find

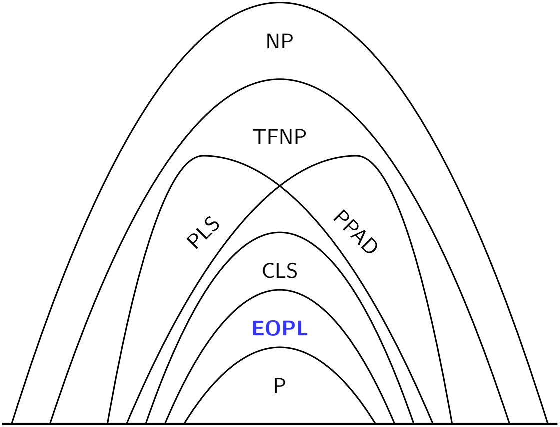
- The end of a line
- A vertex where the potential increases

Unique EOPL



In Unique EOPL, it is promised that the line is **unique**

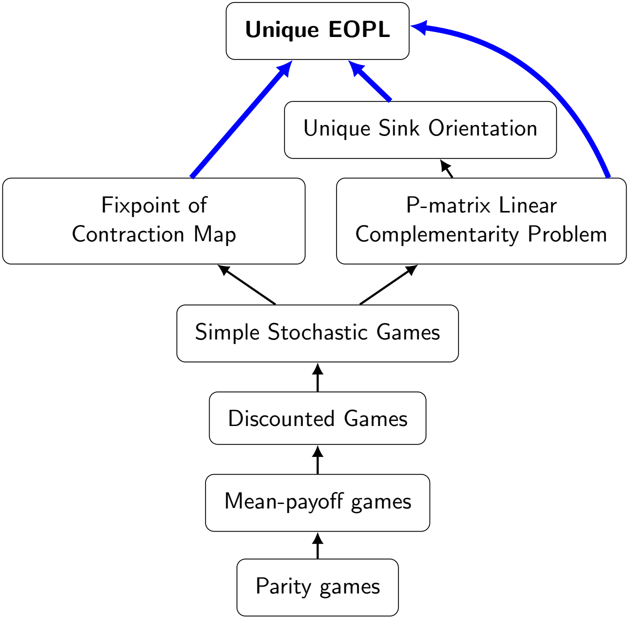
Complexity classes between P and NP



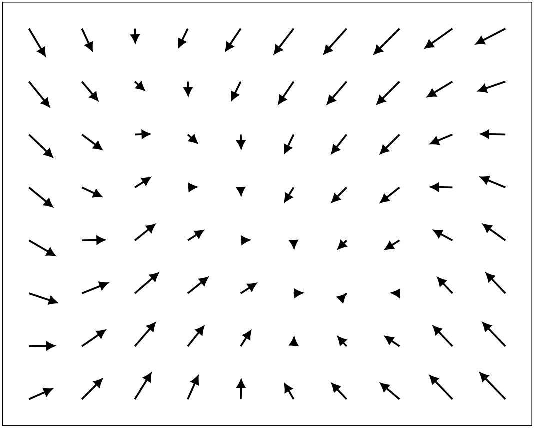
EOPL naturally lies at the intersection of PPAD and PLS

- We also show that it is **contained in CLS**

Our main results



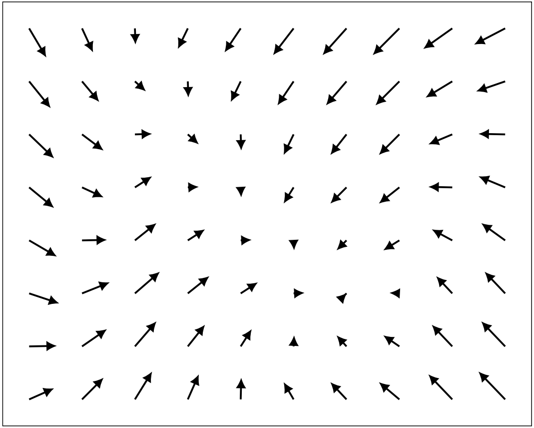
Contraction Maps



f is **contracting** if

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

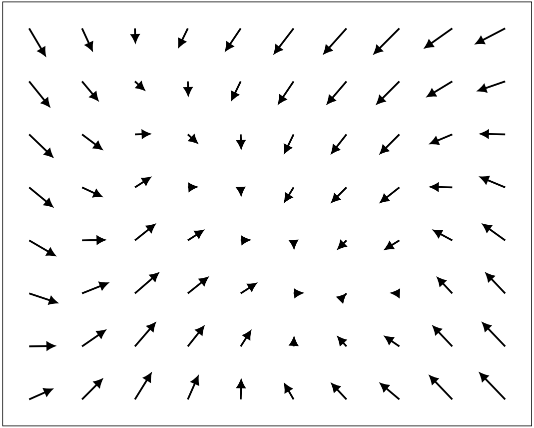
Contraction Maps



Banach's fixpoint theorem

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

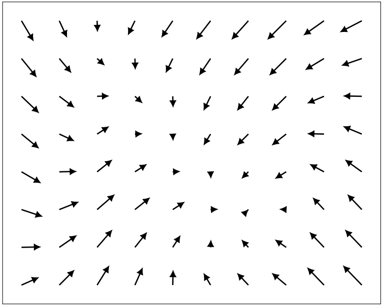
Contraction Maps

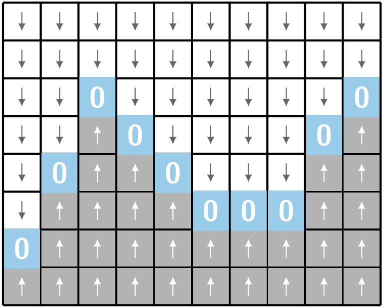


Problem: given a contraction map as an arithmetic circuit

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

Contraction to Unique EOPL

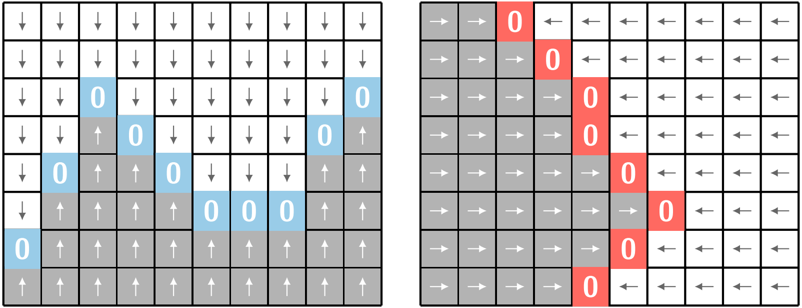




First we discretize the problem

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

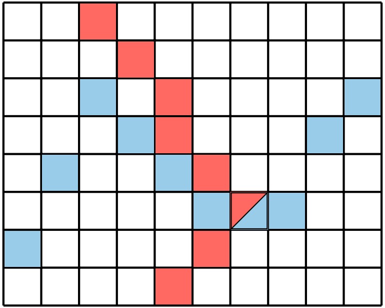
Contraction to Unique EOPL



Discrete contraction

- Find a point that is **0** in all dimensions

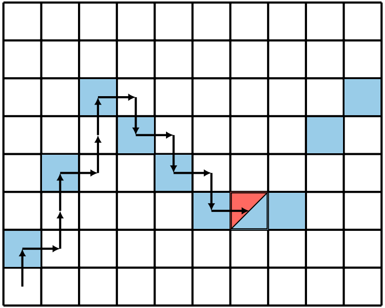
Contraction to Unique EOPL



A point is on the **surface** if it is $\mathbf{0}$ 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

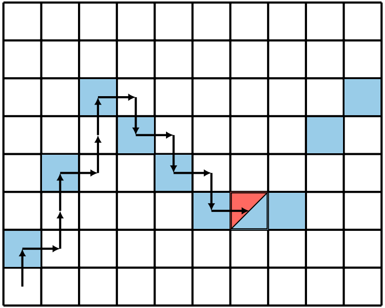
Contraction to Unique EOPL



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

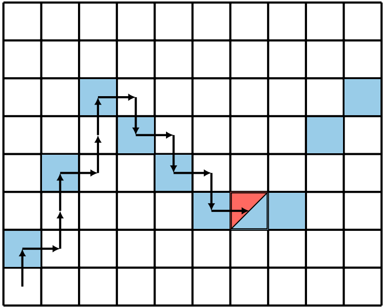
Contraction to Unique EOPL



The potential

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

Contraction to Unique EOPL

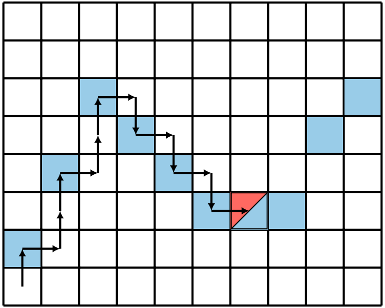


So we can use a pair (a, b) ordered lexicographically where

- 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

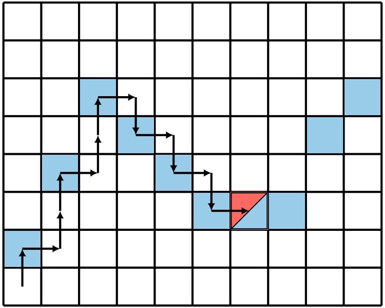
Contraction to Unique EOPL



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

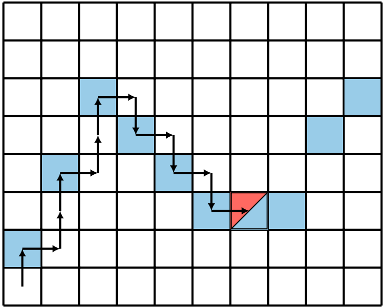
Contraction to Unique EOPL



This generalises to arbitrary dimension

- We walked along the blue surface to reach the red surface

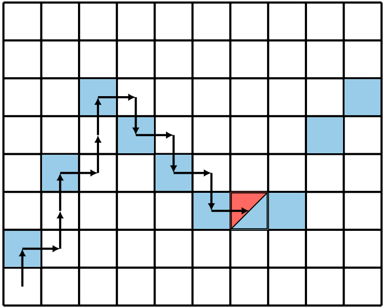
Contraction to Unique EOPL



In 3D

- Walk along the red/blue surface to find the green surface
- Between any two points on the red/blue surface
  - Walk along the blue surface to find the red surface

Contraction to Unique EOPL



Theorem

Contraction is in EOPL, Promise-Contraction is in UniqueEOPL

Consequences for contraction

Theorem

Given an arithmetic circuit C encoding a contraction map

$$f : [0, 1]^d \to [0, 1]^d$$

with respect to any ℓ_p norm

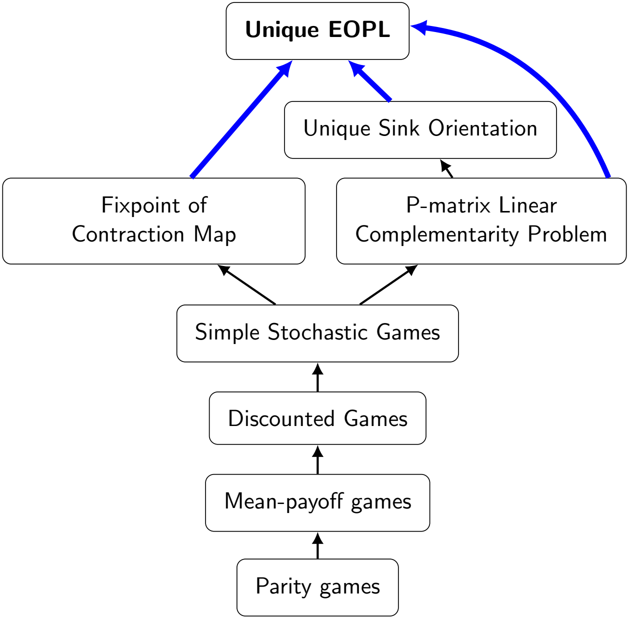
there is an algorithm, based on a **nested binary search**

that finds a fixpoint of f in time

- polynomial in size(C)
- exponential in d

Before, **such algorithms were only known for ℓ_2 and ℓ_∞**

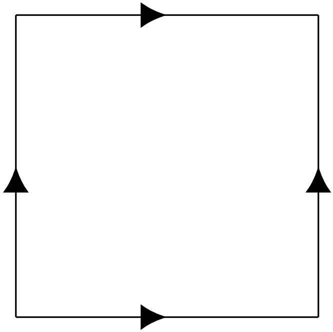
Our main results

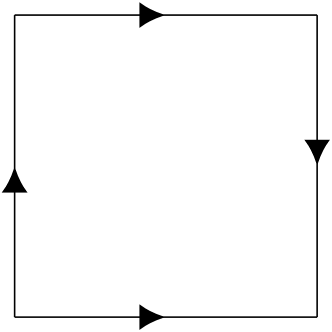


Unique Sink Orientations of Cubes

Orient the edges of an ***n***-dimensional cube

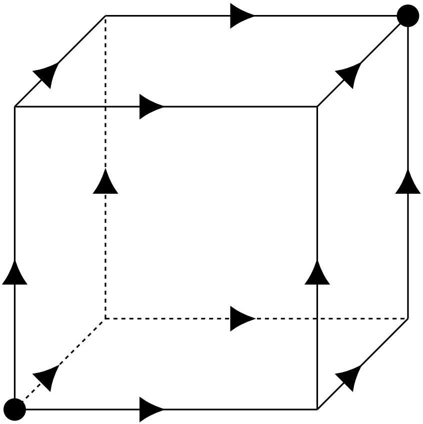
- So that every face has a **unique** sink





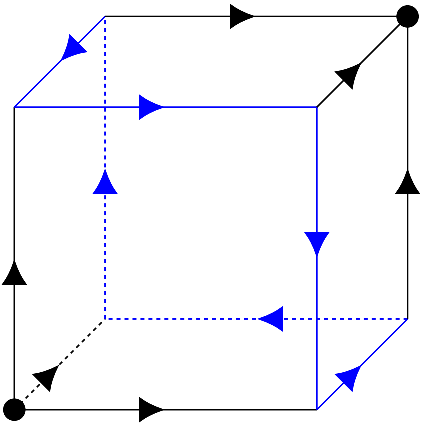
Unique Sink Orientations of Cubes

A 3-dimensional USO



Unique Sink Orientations of Cubes

Can be **cyclic**:



UniqueSinkOrientation

Given a **polynomial-time boolean circuit**

$$\boldsymbol{C : \{0,1\}^n \mapsto \{0,1\}^n}$$

that maps a vertex $\boldsymbol{v}$ of then $\boldsymbol{n}$-cube to the orientation at $\boldsymbol{v}$:

- **find the sink of the cube**
- or a violation to the USO property

Why is USO interesting?

Long line of work on UniqueSinkOrientation:

P-matrix LCP reduces to UniqueSinkOrientation

[Stickney and Watson '78]

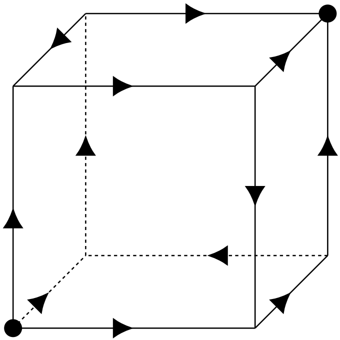
Non-trivial USO algorithms (previously best for P-matrix LCP)

[Szabó and Welzl '01]

Some problems reduce to **acyclic** USO

- parity games
- mean-payoff games
- discounted games
- simple-stochastic games

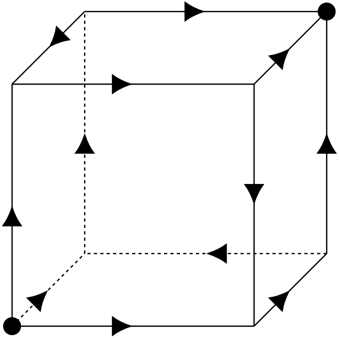
USO in EOPL



Previously

- USO was known to be in TFNP
- But not PPAD or PLS

USO in EOPL

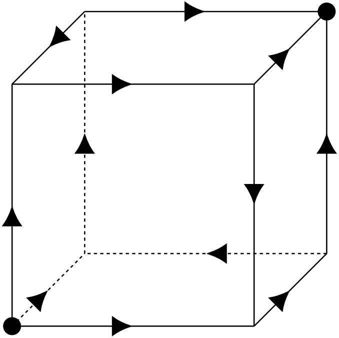


Theorem

USO is in EOPL, Promise-USO is in UniqueEOPL

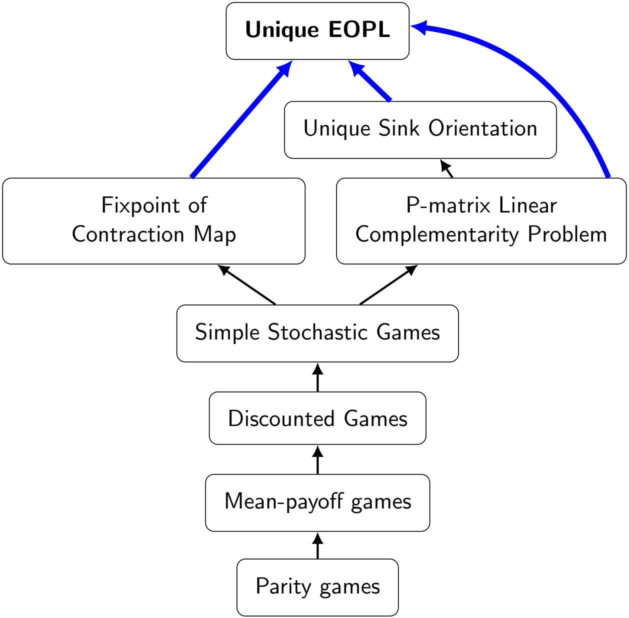
Using similar techniques to Contraction

USO in EOPL

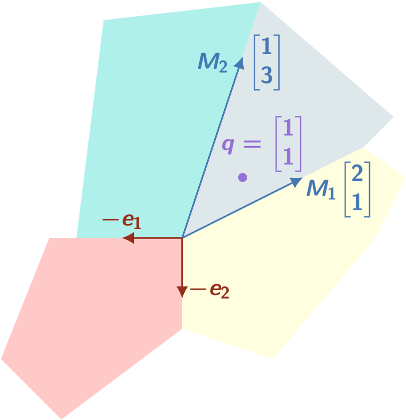


So we put USO in EOPL, CLS, PPAD, and PLS

Our main results



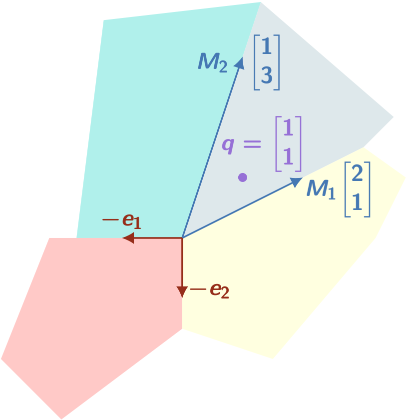
P-matrix Linear Complementarity Problem



Input:

- Vectors $M_1, M_2, \ldots, M_d$
- A vector q

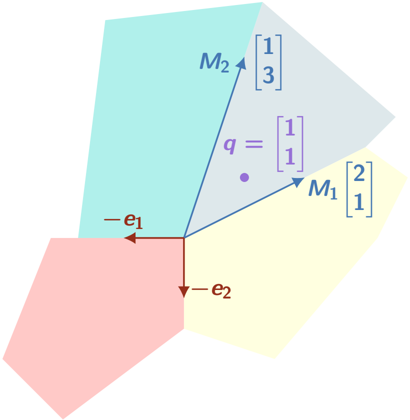
P-matrix Linear Complementarity Problem



A **complementary cone** is all non-negative linear combinations of

- A subset of $\boldsymbol{M_1}$, $\boldsymbol{M_2}$, ..., $\boldsymbol{M_d}$, with
- $-\boldsymbol{e_i}$ in place of each vector not chosen

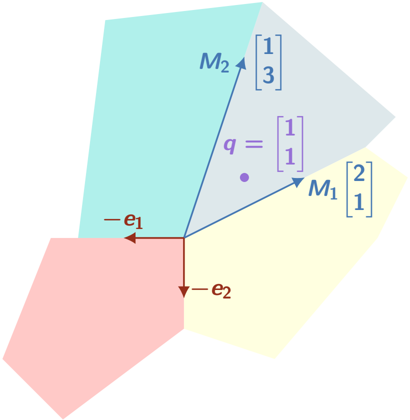
P-matrix Linear Complementarity Problem



The **linear complementarity problem** (LCP)

- Find a cone that contains $\boldsymbol{q}$

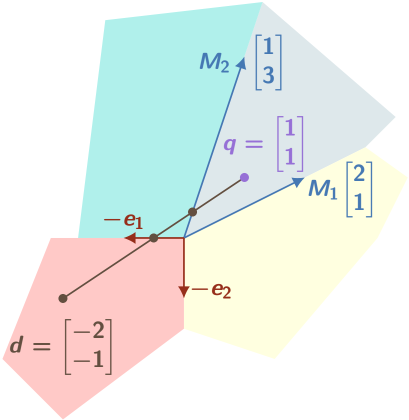
P-matrix Linear Complementarity Problem



P-matrix LCPs

- The cones are guaranteed to exactly partition the space

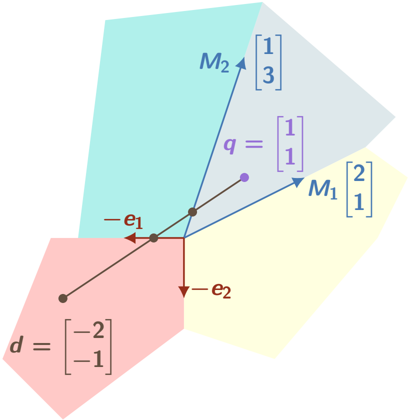
P-matrix Linear Complementarity Problem



We reduce P-matrix LCP to EOPL using **Lemke's algorithm**

- Start at the vector $\boldsymbol{d}$ in the cone $-\boldsymbol{e}_1$, $-\boldsymbol{e}_2$
- Walk through the sequence of cones from $\boldsymbol{d}$ to $\boldsymbol{q}$

P-matrix Linear Complementarity Problem



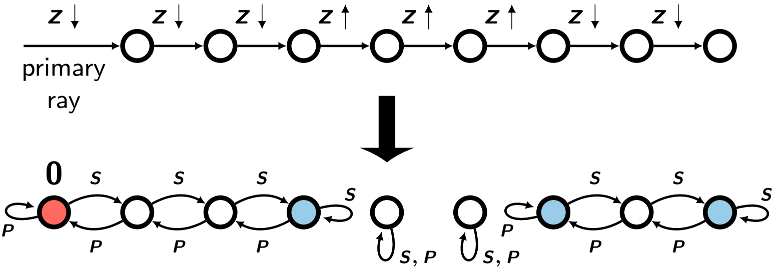
The progress along the path gives us a **potential**

- The algorithm has a variable z
- z corresponds to distance along the path
- it monotonically decreases

P-matrix LCP → EOPL

If the input is not a P-matrix, then z may **increase**

- We deal with this by introducing **new solutions**



P-matrix LCP $\rightarrow$ EOPL

Theorem

P-matrix LCP is in EOPL

Theorem

Promise P-matrix LCP is in Unique EOPL

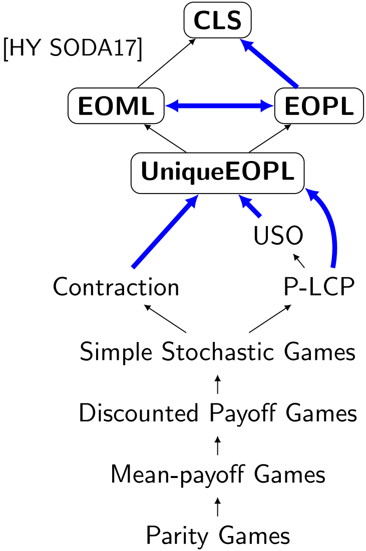
Consequences for P-matrix LCP

Blowup of reduction to EOPL is only **linear**

This allows us to apply an algorithm of **Aldous (1983)**

Gives **fastest-known (randomized) algorithm** for P-matrix LCP, with running time

$$\mathbf{2^{\frac{n}{2}} \cdot poly(n)}$$


CLS
[HY SODA17]
EOML
EOPL
UniqueEOPL
USO
Contraction
P-LCP
Simple Stochastic Games
Discounted Payoff Games
Mean-payoff Games
Parity Games

Conjectures

USO is hard for EOPL
Promise USO is hard for UniqueEOPL

Contraction is hard for EOPL
Promise Contraction is hard for UniqueEOPL

P-matrix LCP is hard for EOPL
Promise P-matrix LCP is hard for UniqueEOPL

Conjectures

CLS $\neq$ UniqueEOPL

CLS =? EOPL

- Could go either way
- If false, which further problems in CLS are also in EOPL?

Thanks!