

01/26/26

More on general-sum games:
Stackelberg leader strategies, Nash
equilibria complexity and algorithms

Your guide:
Avrim Blum

[Readings: Ch. 2.1-2.4 of AGT book]

One more interesting game

“Ultimatum game”:

- Two players “Splitter” and “Chooser”
- 3rd party puts \$10 on table.
- Splitter gets to decide how to split between himself and Chooser.
- Chooser can accept or reject.
- If reject, money is burned.

One more interesting game

“Ultimatum game”: E.g., with \$4

Splitter: how much
to offer chooser

Chooser:
how
much to
accept

	1	2	3
1	(1,3)	(2,2)	(3,1)
2	(0,0)	(2,2)	(3,1)
3	(0,0)	(0,0)	(3,1)

Stackelberg leader strategies

Strategy such that if you announce it and opponent best-responds to you, you are best off.

Splitter: how much to offer chooser

Chooser:
how
much to
accept

	1	2	3
1	(1,3)	(2,2)	(3,1)
2	(0,0)	(2,2)	(3,1)
3	(0,0)	(0,0)	(3,1)

Stackelberg leader strategies

Strategy such that if you announce it and opponent best-responds to you, you are best off.

Need not be a Nash equilibrium.

	Compete	Leave
Price high	(3,3)	(6,1)
Price low	(2,0)	(4,1)

Leader strategy: prob 1/3 on high, 2/3 on low. Think of as $\lim_{\epsilon \rightarrow 0} (1/3 - \epsilon, 2/3 + \epsilon)$

Stackelberg leader strategies

Can solve efficiently. Say we're row player:

- For each column j , solve for p to maximize our expected gain s.t. j is best-response.
- Choose best.

	Compete	Leave
Price high	(3,3)	(6,1)
Price low	(2,0)	(4,1)

Leader strategy: prob 1/3 on high, 2/3 on low. Think of as $\lim_{\epsilon \rightarrow 0} (1/3 - \epsilon, 2/3 + \epsilon)$

Stackelberg leader strategies

Can solve efficiently. Say we're row player:

- For each column j , solve for p to maximize our expected gain s.t. j is best-response.
- Choose best.
 - For each j , solve for $p_1, \dots, p_n \geq 0, \sum_i p_i = 1$, to maximize our gain $\sum_i p_i R_{ij}$ subject to:
 - For each j' , $\sum_i p_i C_{ij} \geq \sum_i p_i C_{ij'}$ (the column player prefers j)

Hardness of computing Nash equilibria

Looking at 2-player n-action games.

2 types of results:

- NP-hardness for NE with special properties
[Gilboa-Zemel] [Conitzer-Sandholm]
 - Is there one with payoff at least v for row?
 - Is there one using row #1?
 - Is there more than one?
 - ...
- PPAD-hardness for finding any NE.
[Chen-Deng][Daskalakis-Goldberg-Papadimitriou]

Hardness of computing Nash equilibria

NP-hardness for NE with special properties

Basic idea:

- Given 3-SAT formula F , create a game with one row for each literal, variable, & clause.
- Also a default attractor action f . $C = R^T$.
- Somehow set things up so that except for (f,f) , all NE must correspond to satisfying assignments.

What about just finding some NE?

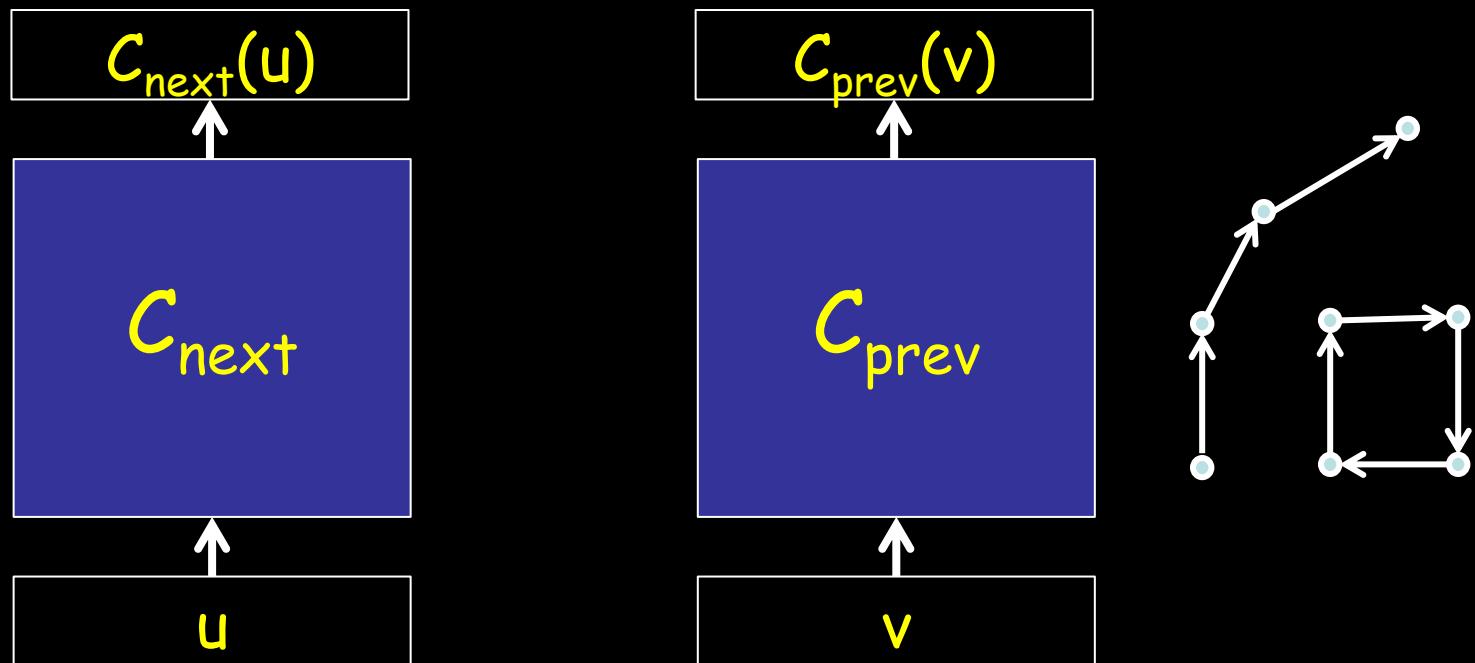
This is "PPAD" hard.

What's that?

What about just finding some NE?

Consider the following problem:

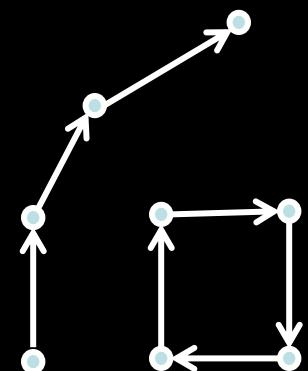
- Given two circuits C_{next} and C_{prev} , each with n -bit input, n -bit output.
- View as defining directed graph G :
 $u \rightarrow v$ iff $C_{\text{next}}(u) = v$ and $C_{\text{prev}}(v) = u$. (indeg ≤ 1 , outdeg ≤ 1)



What about just finding some NE?

Consider the following problem:

- Given two circuits C_{next} and C_{prev} , each with n -bit input, n -bit output.
- View as defining directed graph G :
 $u \rightarrow v$ iff $C_{\text{next}}(u) = v$ and $C_{\text{prev}}(v) = u$. (indeg ≤ 1 , outdeg ≤ 1)
- Say v “unbalanced” if $\text{indeg}(v) \neq \text{outdeg}(v)$.
- If 0^n is unbalanced, then find another unbalanced node. (must exist)



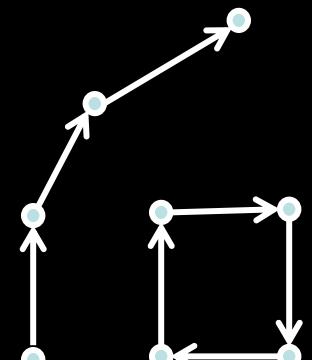
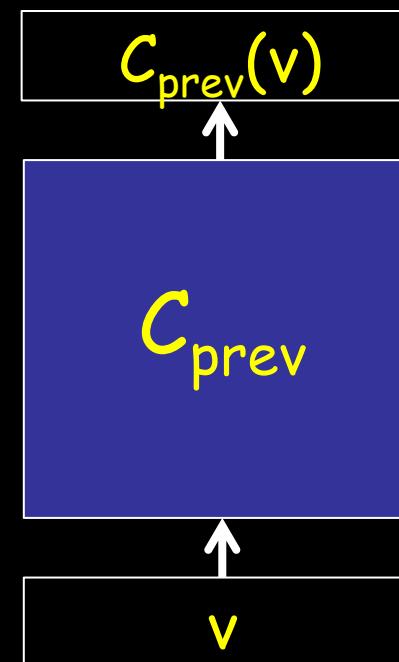
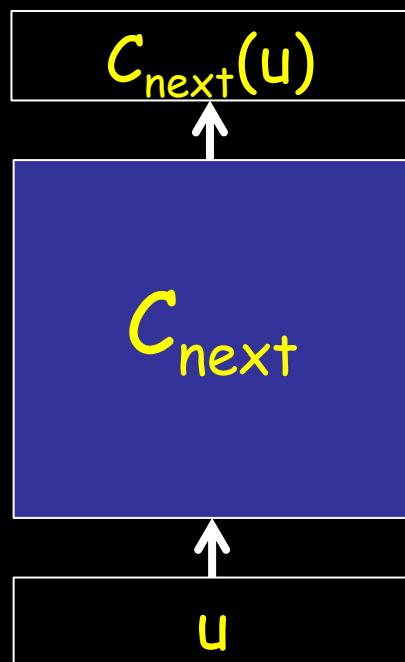
This is PPAD “END OF THE LINE”

What about just finding some NE?

Why isn't this problem trivial? Say $\text{outdeg}(0^n)=1$.

- $\text{for}(u = 0^n; u == C_{\text{prev}}(C_{\text{next}}(u)); u = C_{\text{next}}(u));$

Unfortunately, the path might be exponentially long.

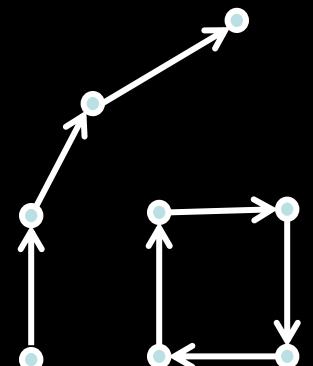


What about just finding some NE?

Not going to give proof that Nash is PPAD-hard.

Instead, give algorithm to show why Nash is in PPAD.

Also another proof of existence of NE



Lemke-Howson algorithm (1964)

Preliminaries: [following discussion in Ch 2]

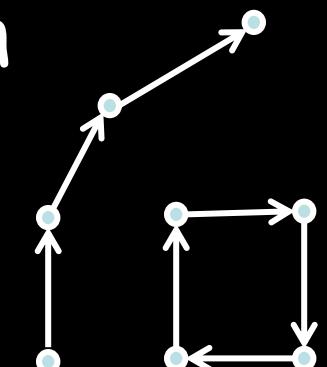
Given: matrices R, C with positive entries.

- For simplicity, convert to symmetric game (A, A^T) : $A =$

0	R
C^T	0

Claim: If $([x,y], [x,y])$ is a symmetric equilib in (A, A^T) , then $(x/X, y/Y)$ is an equilib in (R, C) .

Use $X = \sum_i x_i$, $Y = \sum_i y_i$



Pf: Each player getting payoff $x^T R y + y^T C^T x$ with no incentive to deviate.

Lemke-Howson algorithm (1964)

Given $n \times n$ symmetric game A , find symm equil.

Consider the $2n$ linear constraints on n vars:

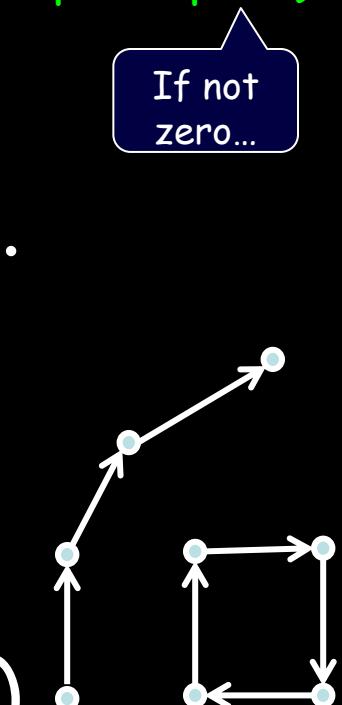
- $A_i z \leq 1$ for all i . $(A_i x \leq 1/Z \text{ where } x_i = z_i/Z)$
- $z_j \geq 0$ for all j . $z = (z_1, z_2, \dots, z_n)$

If not zero...

Assume A is full rank, all A_{ij} non-neg.

- Implies have a bounded polytope.
- And all vertices have n tight constraints (at equality).

Alg will start at the origin (a vertex) and move along edges to a NE.



Lemke-Howson algorithm (1964)

Given $n \times n$ symmetric game A , find symm equil.

Consider the $2n$ linear constraints on n vars:

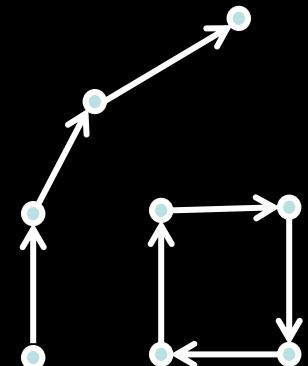
- $A_i z \leq 1$ for all i . $(A_i x \leq 1/Z \text{ where } x_i = z_i/Z)$
- $z_j \geq 0$ for all j . $z = (z_1, z_2, \dots, z_n)$

If not zero...

Strategy i is "represented" if $A_i z = 1$ or $z_i = 0$ (or both)

What if all strategies represented?

- Either $z = (0, \dots, 0)$ or (x, x) is a symmetric Nash.



Lemke-Howson algorithm (1964)

Alg: start at $(0, \dots, 0)$, move along edge.

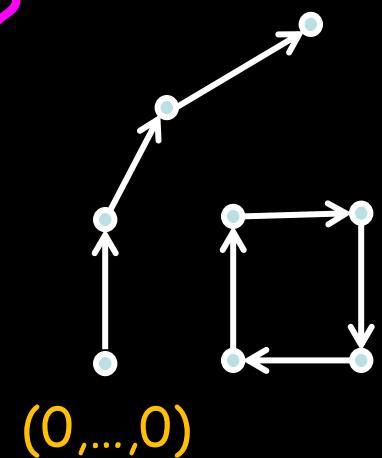
(Relax one of $z_j=0$ and move until hit some $A_i z=1$)

- If $i=j$, then all strategies represented!
- Else i is represented twice.

Strategy i is “represented” if $A_i z=1$ or $z_i=0$ (or both)

What if all strategies represented?

- Either $z=(0, \dots, 0)$ or (x, x) is a symmetric Nash.



Lemke-Howson algorithm (1964)

Alg: start at $(0, \dots, 0)$, move along edge.

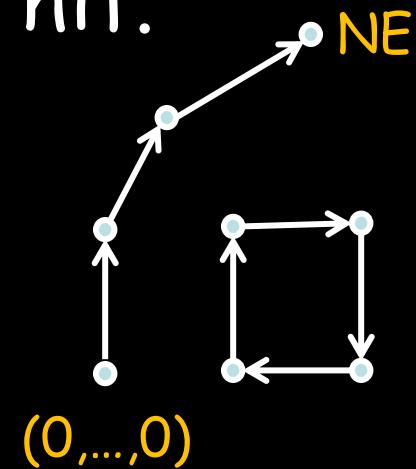
(Relax one of $z_j=0$ and move until hit some $A_i z=1$)

- If $i=j$, then all strategies represented!
- Else i is represented twice.

In general, take strategy represented twice and relax constraint you didn't just hit.

Claim: can't cycle or reach $(0, \dots, 0)$.

End is a Nash equilibrium.



Lemke-Howson algorithm (1964)

Example:

0	0	0
---	---	---

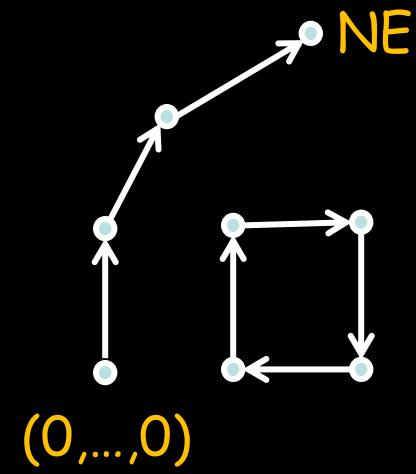
*		
-	0	0

*	*	
-	0	-

*	*	
-	-	0

*		
0	-	0

*		
-	0	0
0	0	0



Lemke-Howson algorithm (1964)

One implication: every non-degenerate game has an odd number of Nash equilibria.

