Minmax Theorem and Lemke-Howson Algorithm

CS 580

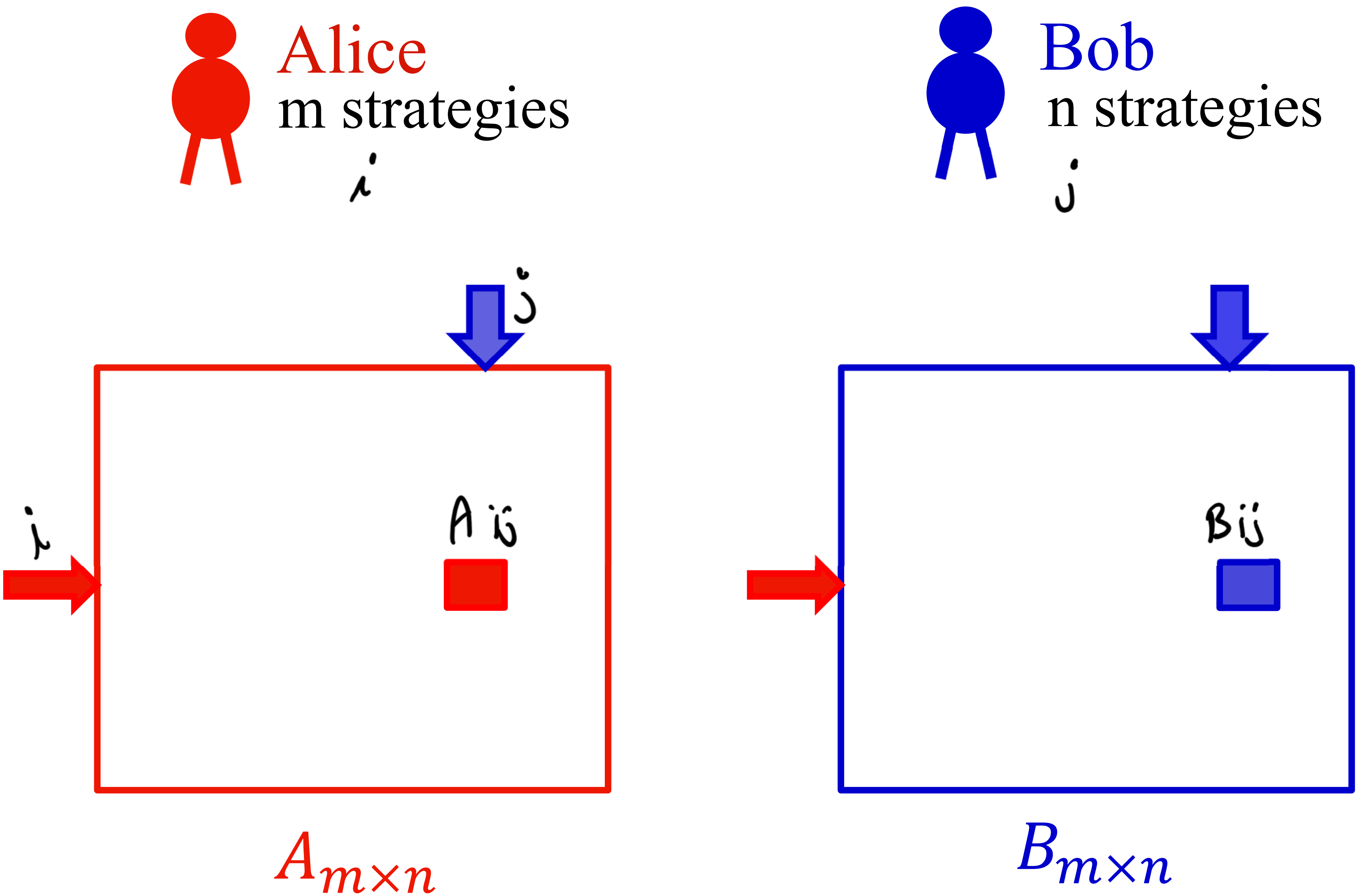
Instructor: Ruta Mehta

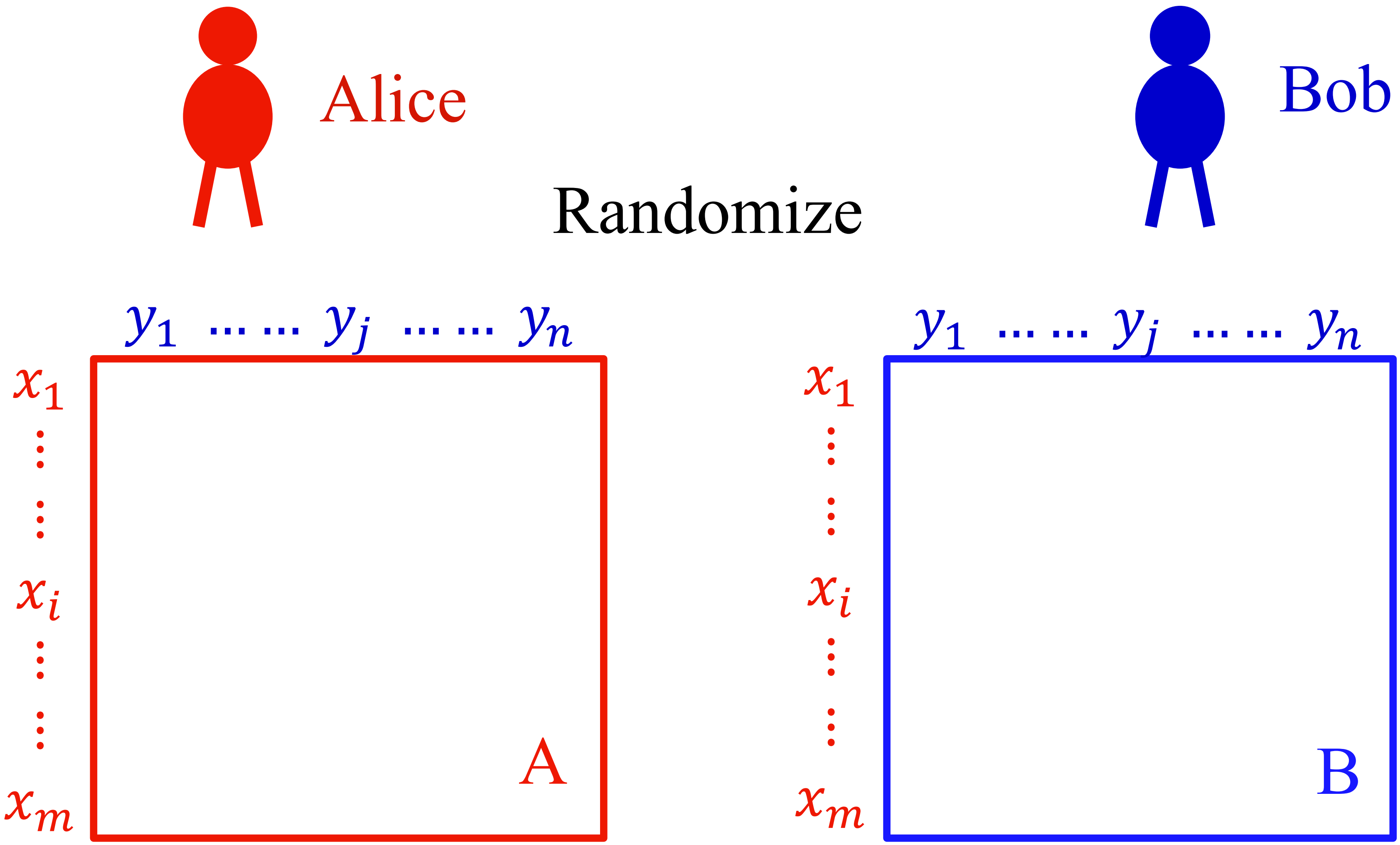
ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Agenda

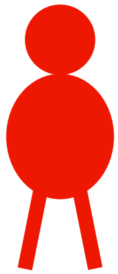
- Two-player Games, NE (recall)
- Zero-sum games
  - Minmax Theorem
  - LP-duality
- Lemke-Howson Algorithm
- Class PPAD

Our focus: Two-player games




Alice
Bob
Randomize
y_1 …… y_j …… y_n
x_1
x_i
x_m
A
y_1 …… y_j …… y_n
x_1
x_i
x_m
B

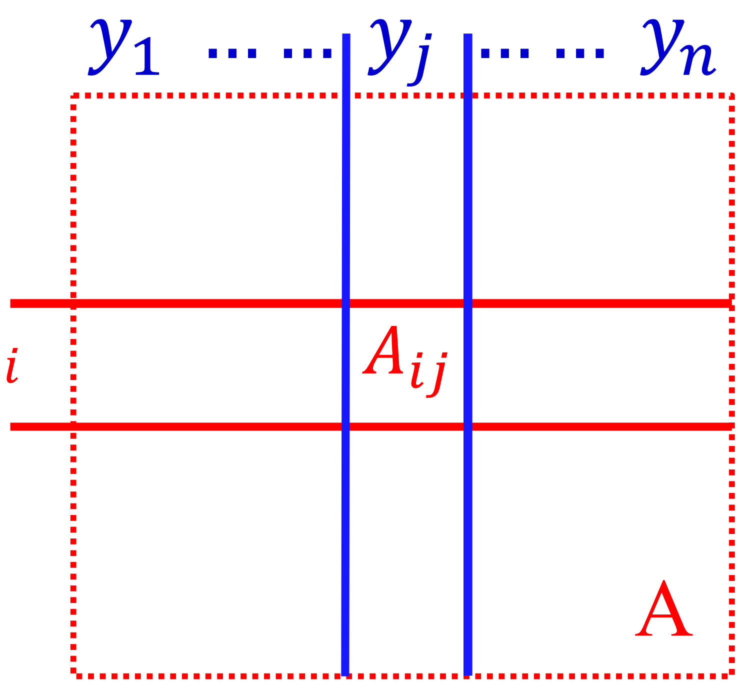
2-Nash Characterization



Alice

Bob

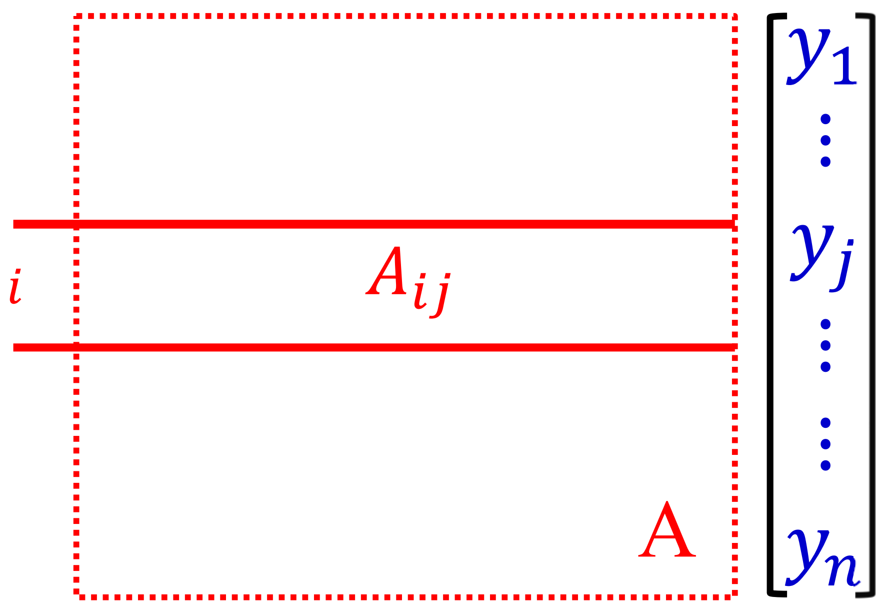
- For Alice, i^{th} strategy gives



$$\longrightarrow \sum_j A_{ij} y_j$$

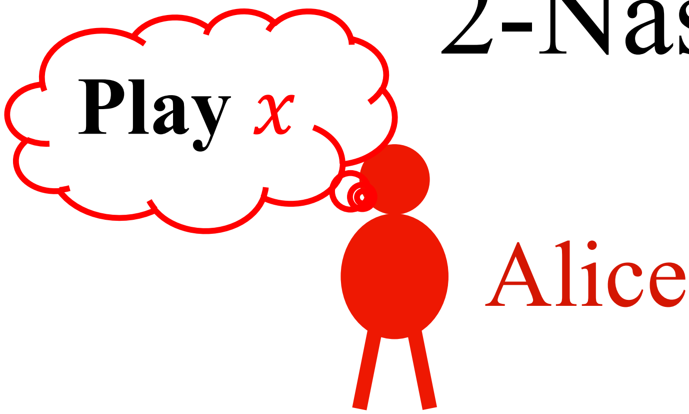
2-Nash Characterization

- For Alice, i^{th} strategy gives

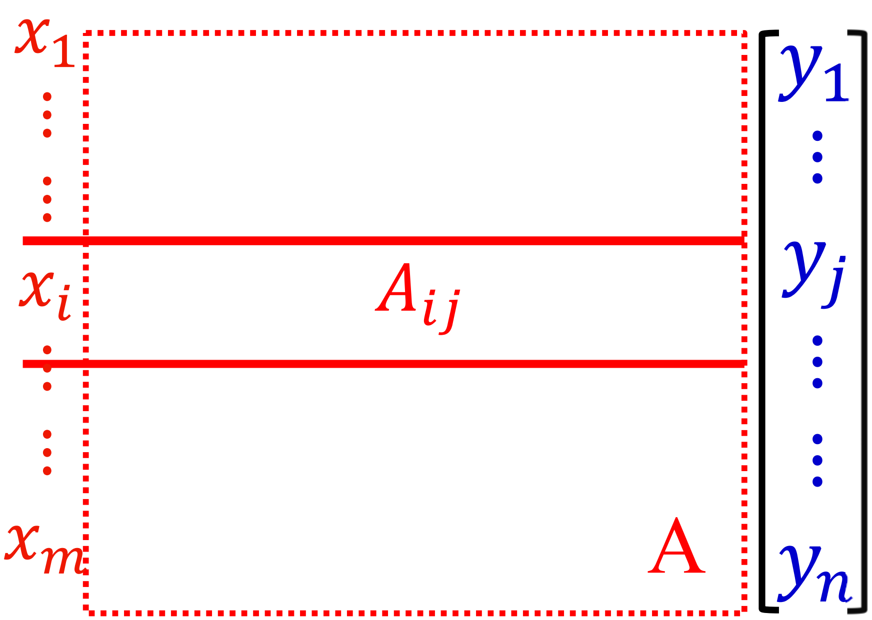


$$\rightarrow \sum_j A_{ij} y_j = (Ay)_i$$

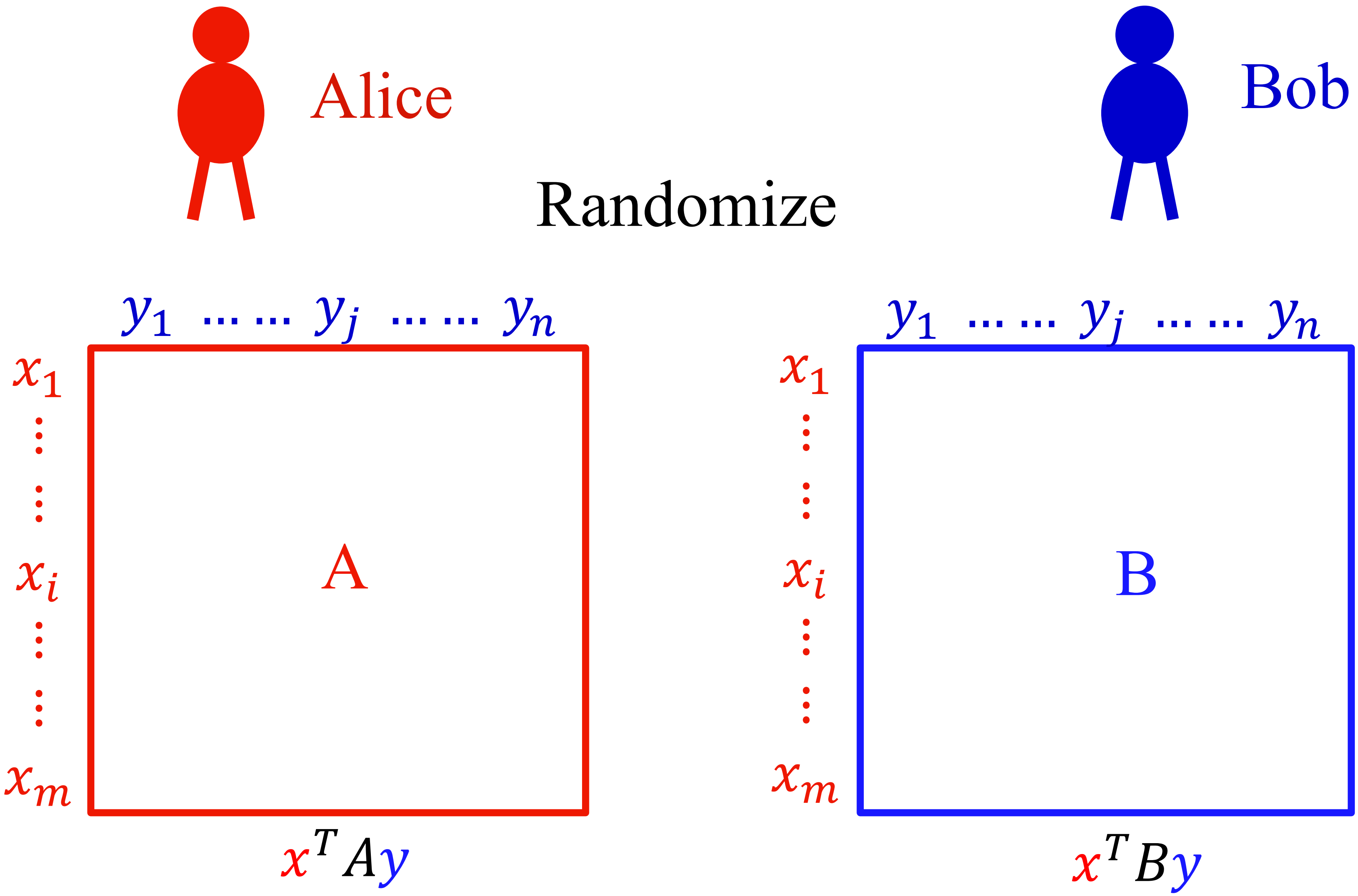
2-Nash Characterization



- Alice's expected payoff is



$$\sum_i x_i (Ay)_i = x^T A y$$



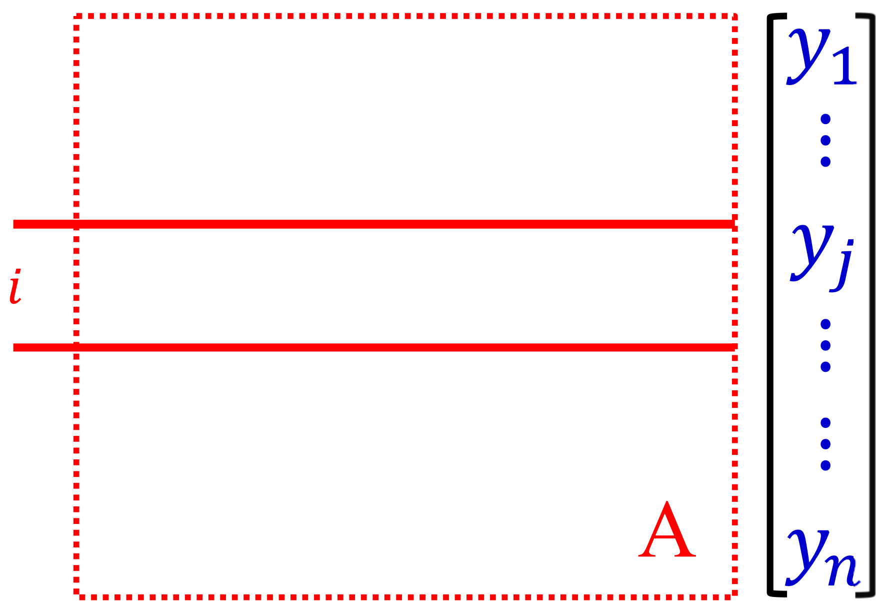
NE: No unilateral deviation is beneficial

$$x^T A y \geq z^T A y, \qquad \forall z \in \Delta_m$$
$$x^T B y \geq x^T B z, \qquad \forall z \in \Delta_n$$

Nash Eq. Characterization

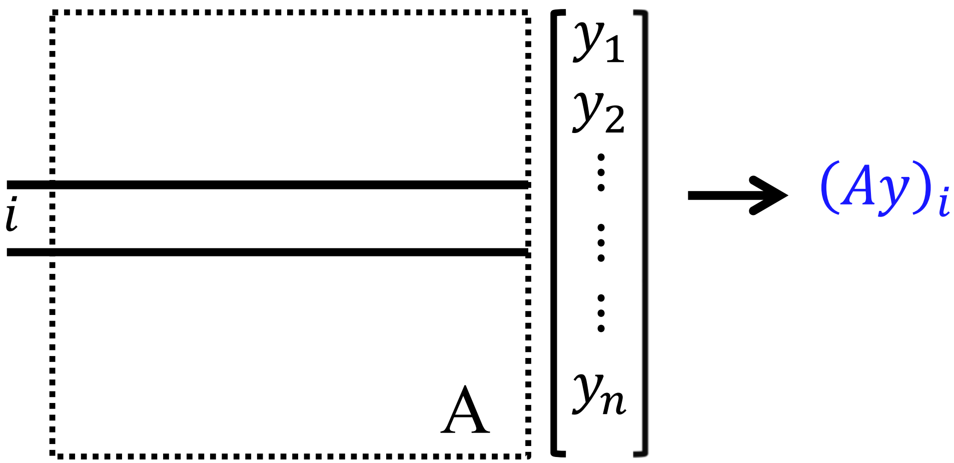
2-Nash Characterization

- For Alice, i^{th} strategy gives



$$\rightarrow \sum_j A_{ij} y_j = (Ay)_i$$

- i^{th} strategy gives Alice



- Max possible payoff: $\max_i (Ay)_i$ $\geq x^T A y \quad \forall x \in \Delta_m$

$=$

- x achieves max payoff iff

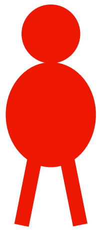
$$x^T A y \geq (Ay)_i, \qquad \forall i$$

$$\equiv$$

$$\forall k, \qquad x_k > 0 \Rightarrow k \in \operatorname*{argmax}_i (Ay)_i$$

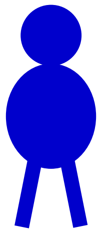
Complementarity

Polyhedra



max-payoff $\leq \pi_A$

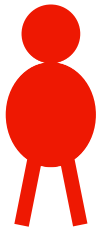
$$P \quad \boxed{\begin{array}{c} \forall i, (Ay)_i \leq \pi_A \\ y \in \Delta_n \end{array}}$$

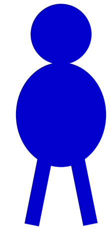


max-payoff $\leq \pi_B$

$$Q \quad \boxed{\begin{array}{c} \forall j, (x^T B)_j \leq \pi_B \\ x \in \Delta_m \end{array}}$$

NE Characterization





max-payoff $\leq \pi_A$

$$P \quad \boxed{\begin{array}{c} \forall i, (Ay)_i \leq \pi_A \\ y \in \Delta_n \end{array}}$$

max-payoff $\leq \pi_B$

$$Q \quad \boxed{\begin{array}{c} \forall j, (x^T B)_j \leq \pi_B \\ x \in \Delta_m \end{array}}$$

NE iff Complementarity

$$\forall i \leq m, \qquad x_i > 0 \Rightarrow (Ay)_i = \pi_A$$

$$\forall j \leq n, \qquad y_j > 0 \Rightarrow (x^T B)_j = \pi_B$$

$$P \quad \boxed{\begin{array}{c} \forall i, (Ay)_i \le \pi_A \\ y \in \Delta_n \end{array}} \qquad\qquad Q \quad \boxed{\begin{array}{c} \forall j, (x^T B)_j \le \pi_B \\ x \in \Delta_m \end{array}}$$

$$(y, \pi_A) \in P, \qquad (x, \pi_B) \in Q$$

2-Nash

$$\text{max: } \boxed{x^T(A+B)y - (\pi_A + \pi_B)}$$

≤ 0

$= 0$ iff π_A, π_B are exactly the max payoffs & (x, y) is a NE

$$\text{s.t. } (y, \pi_A) \in P, (x, \pi_B) \in Q$$

Zero-sum Games

Von Neuman’s maxmin theorem (1928) = LP-duality

P $\boxed{\begin{array}{c} \forall i, (Ay)_i \leq \pi_A \\ y \in \Delta_n \end{array}}$ Q $\boxed{\begin{array}{c} \forall j, (x^T B)_j \leq \pi_B \\ x \in \Delta_m \end{array}}$

$$(y, \pi_A) \in P, \qquad (x, \pi_B) \in Q$$

Theorem. If (A, B) is zero-sum, i.e., $A + B = 0$, then 2-Nash → linear programming

$$\text{max: } x^T(A+B)y - (\pi_A + \pi_B)$$

$$\text{s.t. } (y, \pi_A) \in P, \qquad (x, \pi_B) \in Q$$

$$P \quad \boxed{\begin{array}{c} \forall i, (Ay)_i \le \pi_A \\ y \in \Delta_n \end{array}} \qquad\qquad Q \quad \boxed{\begin{array}{c} \forall j, (x^T B)_j \le \pi_B \\ x \in \Delta_m \end{array}}$$

$$(y, \pi_A) \in P, \qquad (x, \pi_B) \in Q$$

Theorem. If (A, B) is zero-sum, i.e., $A + B = 0$, then

2-Nash → linear programming

$$\text{max: } -(\pi_A + \pi_B)$$

$$\text{s.t. } (y, \pi_A) \in P, \qquad (x, \pi_B) \in Q$$

Theorem. [von Neumann'28] (max-min = min-max) Game $(\cancel{A,-A})$.
Wrt $\boxed{A}$, Alice is a maximizer and Bob minimizer. Then,

1. Alice
2. Bob

$$\boxed{\max_x \min_y x^T A y} \overset{\le}{=} \boxed{\min_y \max_x x^T A y}$$ & the max-min is NE.

1. Bob
2. Alice

$\rightarrow$ It does not matter who goes first.

① $x^* = \arg\max_x \min_y x^T A y$

② $y^* = \arg\min_y \max_x x^T A y$

Pf: $(\le)$

$$\max_x \min_y x^T A y \underset{(\because ①)}{=} \min_y x^{*T} A y \le x^{*T} A y^* \le \max_x x^T A y^* \underset{(\because ②)}{=} \min_y \max_x x^T A y$$

$(\ge)$ Let $(\tilde{x}, \tilde{y})$ is a NE.

$$\min_y \max_x x^T A y \underset{(\because ②)}{\le} \max_x x^T A \tilde{y} \underset{(NE)}{=} \tilde{x}^T A \tilde{y} = \min_y \tilde{x}^T A y \underset{(\because ①)}{\le} \max_x \min_y x^T A y$$

∎

Consequences.

Value of the game $= \max_x \min_y x^T A y = \min_y \max_x x^T A y$

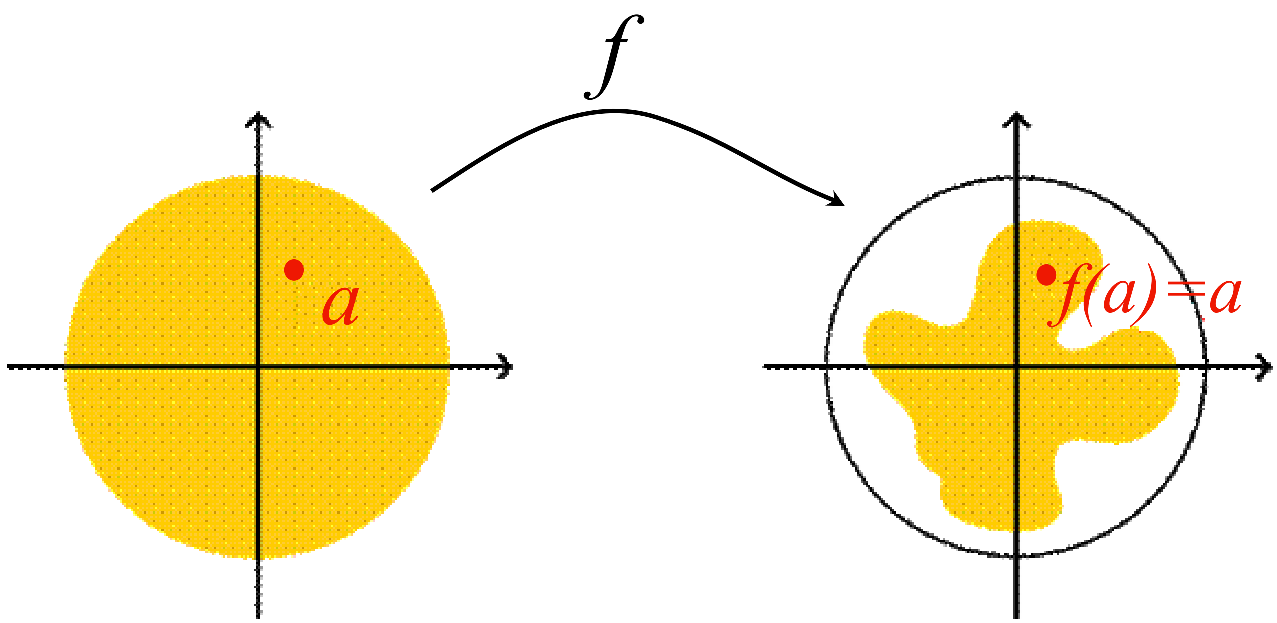
$$\max_x \min_y x^T A y = \tilde{x}^T A \tilde{y} = \min_y \max_x x^T A y \quad \forall (\tilde{x}, \tilde{y}) \text{ NE.}$$

→ LP-duality ⇒ convex set of NE.

→ No matter who goes first the play ends up at NE.

Computation in general?

NE existence via fixed-point theorem.



Computation? (in Econ)

- Special cases: Dantzig'51, Lemke-Howson'64, Elzen-Talman'88, Govindan-Wilson'03, …
- Scarf'67: Approximate fixed-point.
  - Numerical instability
  - Not efficient!
- …

Lemke-Howson (1964)

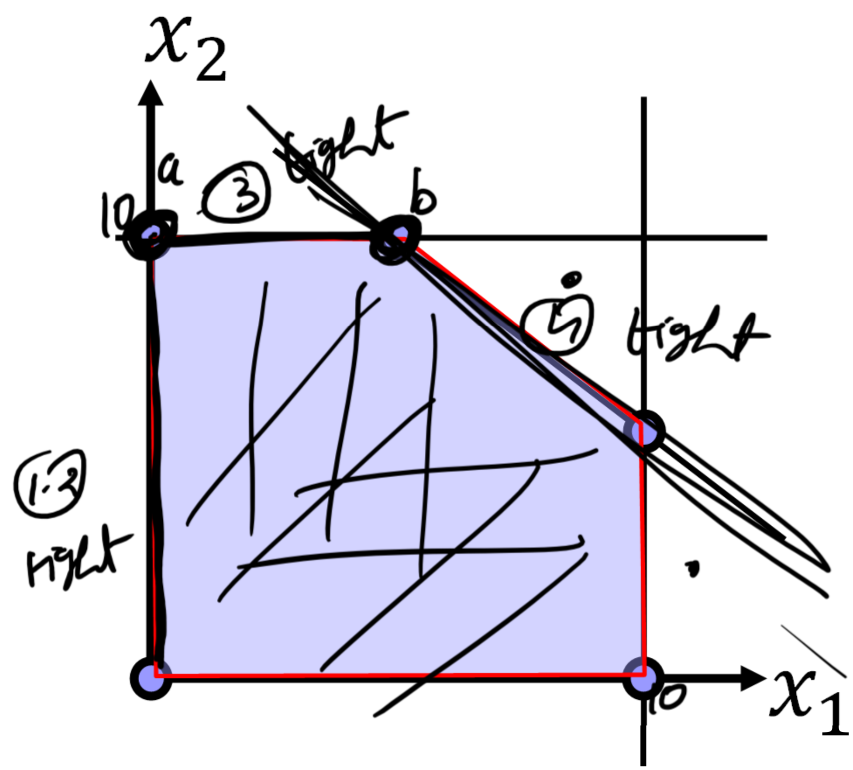
(also a motivation for class PPAD)

Follows a path on a polytope

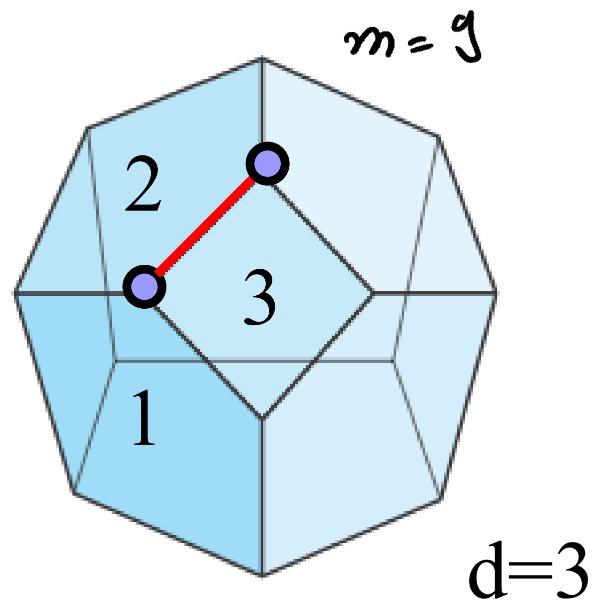
Basic Polytope Properties

Linear inequalities: (**d**imension=2)

$$\text{1.1 } x_1, \overset{1.2}{x_2} \geq 0$$
$$\text{2 } x_1 \leq 10$$
$$\text{3 } x_2 \leq 10$$
$$\text{4 } x_1 + x_2 \leq 15$$



Basic Polytope Properties



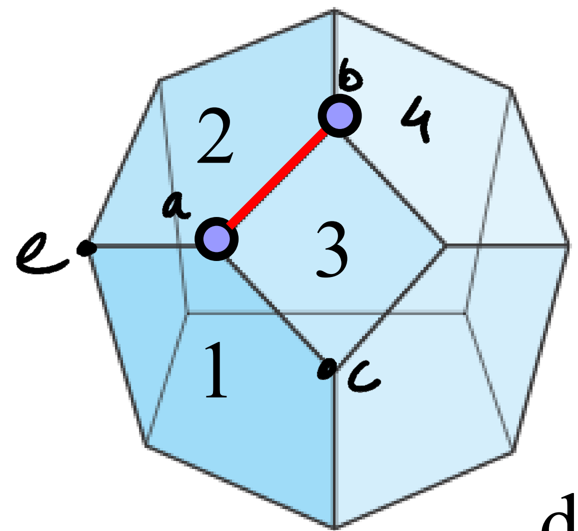
- Given $A_{m\times d}, b_{m\times 1}$:
- $(Ax)_i \ \le b_i, \forall i = 1, \dots, m$
  - In d dimension
  - Hyper-plane i: $(Ax)_i = b_i$

  (Inequality i is *tight* (holds with equality)).
- $(d-k)$ tight inequ. on k-dimensional face
  - At a vertex (0-dim), d tight ineq.
  - On an edge (1-dim), (d-1) tight ineq.

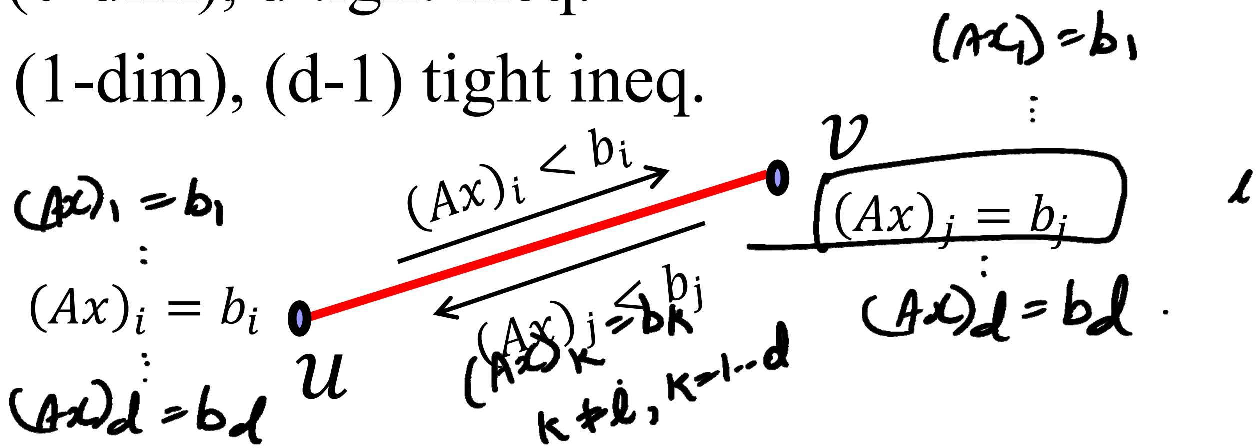
Basic Polytope Properties

- $(Ax)_i \le b_i, \forall i = 1, \ldots, m$
  - □ Hyper-plane i: $(Ax)_i = b_i$

    (Inequality i is *tight* (equality)).
- $(d-k)$ tight inequ. on k-face
  - □ At a vertex (0-dim), d tight ineq.
  - □ On an edge (1-dim), (d-1) tight ineq.



d=3



u, v share d-1 tight ineq.

These are also tight on connecting edge

Finding NE in game (A, B)

$\Downarrow$

NE Characterization

max-payoff $\leq \pi_A$

$$P \quad \begin{array}{c} \forall i, (Ay)_i \leq \pi_A \\ y \in \Delta_n \end{array}$$

max-payoff $\leq \pi_B$

$$Q \quad \begin{array}{c} \forall j, (x^T B)_j \leq \pi_B \\ x \in \Delta_m \end{array}$$

NE iff Complementarity

$$\forall i \leq m, \qquad x_i > 0 \Rightarrow (Ay)_i = \pi_A$$

$$\forall j \leq n, \qquad y_j > 0 \Rightarrow (x^T B)_j = \pi_B$$

NE Characterization

max-payoff $\leq \pi_A$

P $\boxed{\begin{array}{c}\forall i, (Ay)_i \leq \pi_A \\ y \in \Delta_n\end{array}}$

max-payoff $\leq \pi_B$

Q $\boxed{\begin{array}{c}\forall j, (x^T B)_j \leq \pi_B \\ x \in \Delta_m\end{array}}$

NE iff Complementarity

$$\forall i \leq m, \qquad x_i = 0 \text{ or } (Ay)_i = \pi_A$$
$$\forall j \leq n, \qquad y_j = 0 \text{ or } (x^T B)_j = \pi_B$$

Finding NE in game (A, B)

NE iff Complementarity

$$\forall i \leq m, \qquad x_i > 0 \Rightarrow (Ay)_i = \pi_A$$
$$\forall j \leq n, \qquad y_j > 0 \Rightarrow (x^T B)_j = \pi_B$$

Finding NE in game (A, B)

$\Downarrow$

$d = m + n$

Given $M_{d \times d} > 0$, find $x \in R^d$, $x \neq \mathbf{0}$ s.t.

$\forall i \leq d,\ x_i \geq 0, \qquad (Mx)_i \leq 1$

$x_i > 0 \Rightarrow (Mx)_i = 1$

$\equiv x_i = 0$ OR $(Mx)_i = 1$

$M = \begin{bmatrix} 0 & A \\ B^T & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$

Find $x \neq \mathbf{0}$ s.t.

$\forall i \leq d,\ \ x_i \geq 0, \qquad (Mx)_i \leq 1$ → d-dim polytope P

$x_i = 0$ or $(Mx)_i = 1$ ⟶ Label/color i is present

$x_1 = 0$ or $(Mx)_1 = 1$ $\quad$ $x_4 = 0$ or $(Mx)_4 = 1$

$x_2 = 0$ or $(Mx)_2 = 1$ $\quad \ddots$

$x_3 = 0$ or $(Mx)_3 = 1$ $\quad$ $x_d = 0$ or $(Mx)_d = 1$

- Define $L(x) = \{i \mid \text{label/color } i \text{ is present at } x\}$
- *Fully-labeled/panchromatic* set of points

  $S = \{x \mid L(x) = \{1, \ldots, d\}\}.$
  - Vertices.
  - $\mathbf{0} \in S$. $x \in S \setminus \{\mathbf{0}\}$ iff x is a solution → **new goal!**

$x_i = 0$ or $(Mx)_i = 1$ $\longrightarrow$ Label/color i present

- Define $L(x) = \{i \mid \text{label/color } i \text{ is present at } x\}$
- *Fully-labeled* set $S = \{x \mid L(x) = \{1, \ldots, d\}\}$.
  - Vertices.
  - $\mathbf{0} \in S$. $x \in S \setminus \{\mathbf{0}\}$ iff x is a solution → **new goal!**

- *1-almost* fully-labeled set, $S_1 = \{x \mid L(x) \supseteq \{2, \ldots, d\}\}$.
  - $S \subset S_1$. Vertices + edge.

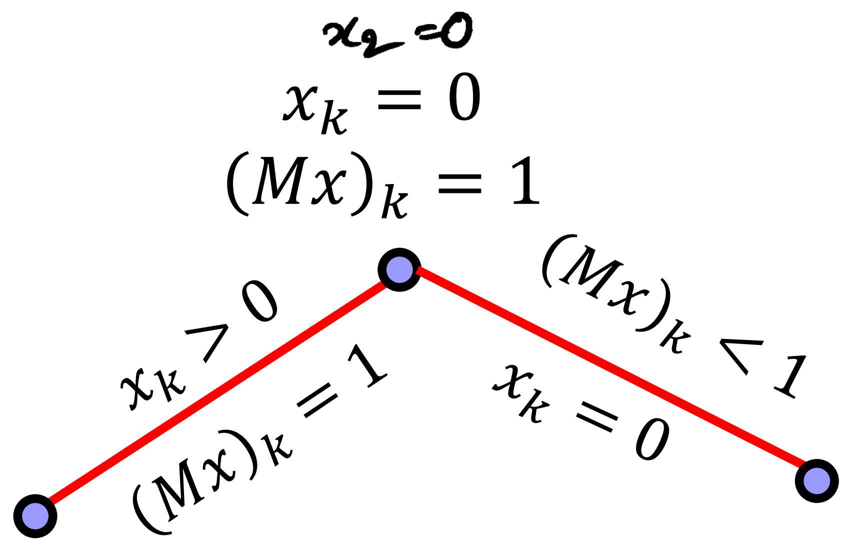
Lemke-Howson follows a path in S_1

Structure of S_1
(Paths and Cycles)

$\forall i, x_i \geq 0, \quad (Mx)_i \leq 1 \rightarrow$ d-dim P

$x_i = 0$ or $(Mx)_i = 1 \longrightarrow$ Label/color i

- Vertex $v \in S_1 \setminus S$. Then $L(v) = \{2, \dots, d\}$
  - For each $i \in \{2, \dots, d\}$, $x_i = 0$ *or* $(Mx)_i = 1$
  - Unique k $\in \{2, \dots, d\}$ s.t. $x_k = 0$ **and** $(Mx)_k = 1$
  - k is duplicate



Both edges are in S_1

Any other? No!

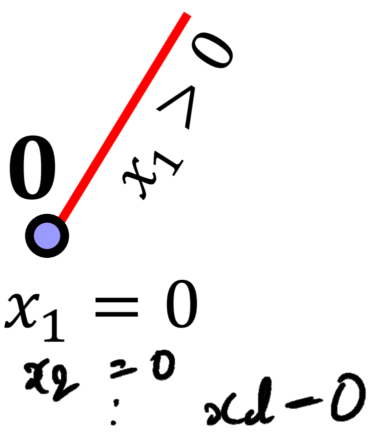
Claim 1. $\deg(v) = 2$ if $v \in S_1 \setminus S$

Starting vertex

$\forall i, x_i \geq 0, \qquad (Mx)_i \leq 1 \rightarrow$ d-dim P

$x_i = 0$ or $(Mx)_i = 1 \longrightarrow$ Label/color i

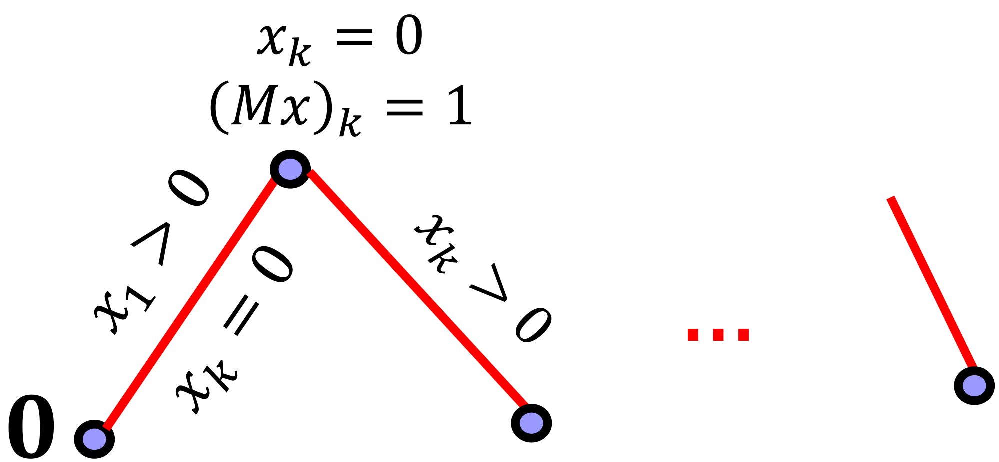
- Vertex $v \in S(\subset S_1)$. Then $L(v) = \{1, \dots, d\}$
  - No duplicate label.
    - Can only leave label 1 to remain in S_1



Claim 2. $\deg(v) = 1$ if $v \in S$

Lemke-Howson: Follow path starting at $\mathbf{0}$

- Vertex $v \in S(\subset S_1)$. Then $L(v) = \{1, \dots, d\}$
  - No duplicate label



Thumb rule: Relax the one that is tight on the previous edge.

1. Leave label 1
2. If Label 1 found
   - Then done.
3. Else leave duplicate label.
4. Go to 2.

Recall

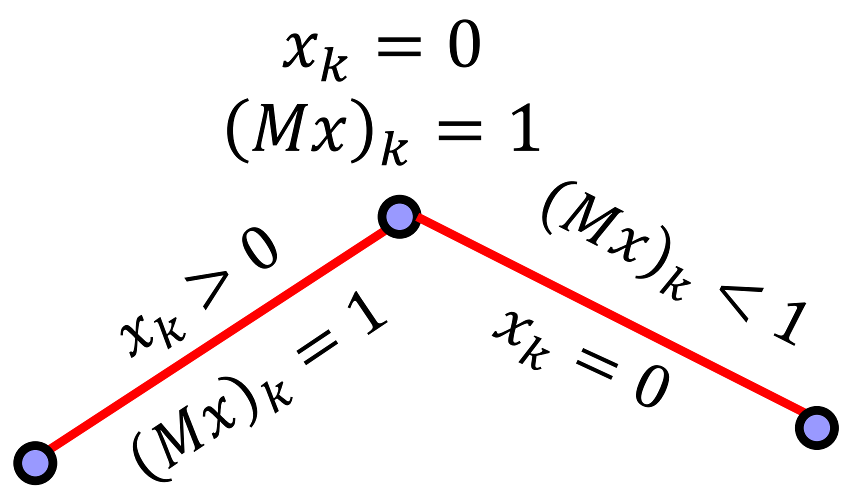
$\forall i, x_i \geq 0,\qquad (Mx)_i \leq 1$ → d-dim P

$x_i = 0$ or $(Mx)_i = 1$ → Label/color i

- Vertex $v \in S_1 \setminus S$. Then $L(v) = \{2, \dots, d\}$
  - For each $i \in \{2, \dots, d\}$, $x_i = 0\ or\ (Mx)_i = 1$
  - Unique k $\in \{2, \dots, d\}$ s.t. $x_k = 0$ **and** $(Mx)_k = 1$
  - k is duplicate

Both edges are in S_1

Any other? No!



Claim 1. $\deg(v) = 2$ if $v \in S_1 \setminus S$

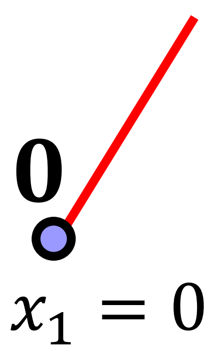
Recall

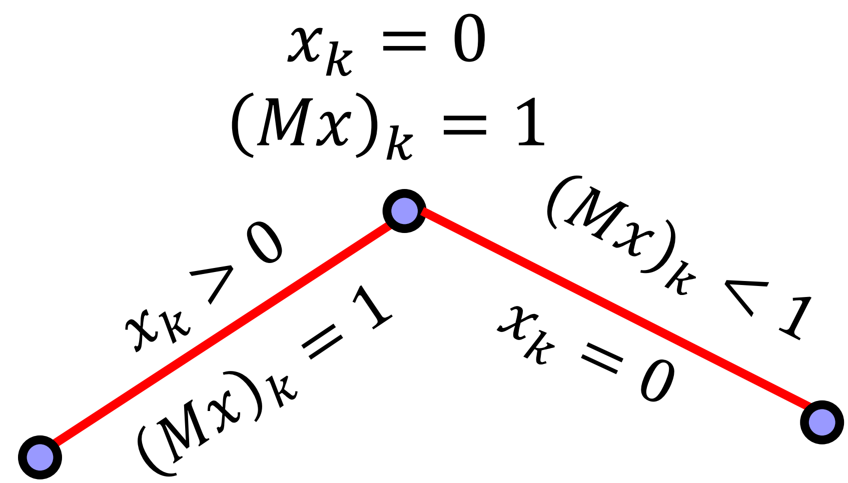
$\forall i, x_i \geq 0, \quad (Mx)_i \leq 1 \rightarrow$ d-dim P

$x_i = 0$ or $(Mx)_i = 1 \longrightarrow$ Label/color i

- Vertex $v \in S_1 \setminus S$. Then $L(v) = \{2, \dots, d\}$

Claim 1. $\deg(v) = 2$ if $v \in S_1 \setminus S$





- Vertex $v \in S (\subset S_1)$. Then $L(v) = \{1, \dots, d\}$
  - No duplicate label.

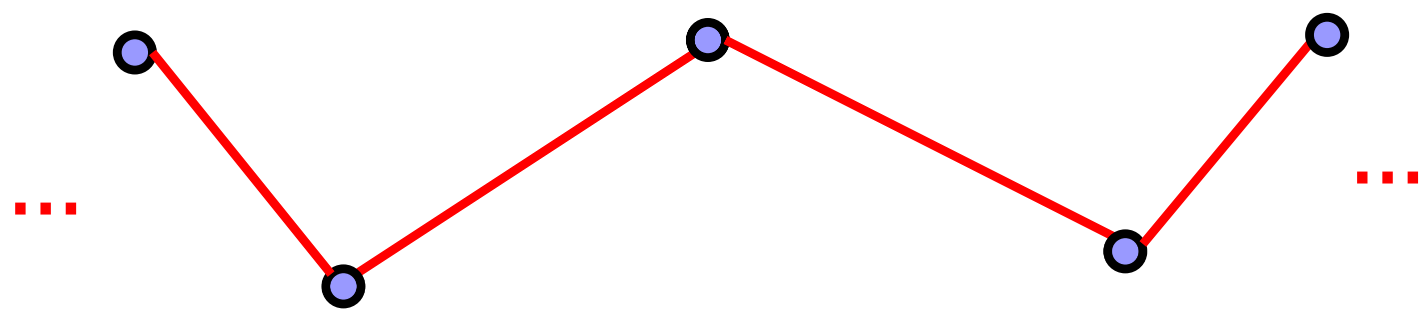
Claim 2. $\deg(v) = 1$ if $v \in S$

S_1: Structure

$\forall i, x_i \geq 0, \quad (Mx)_i \leq 1$ → d-dim P

$x_i = 0$ or $(Mx)_i = 1$ → Label/color i

- Vertex $v \in S_1 \setminus S$. Then $L(v) = \{2, \ldots, d\}$
  - Unique duplicate label



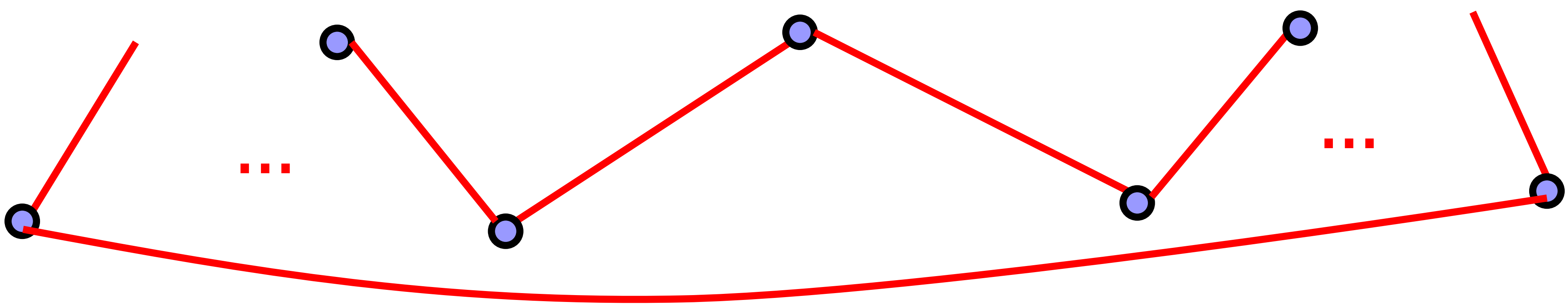
Both edges are in S_1

S_1: Structure

$\forall i, x_i \geq 0, \quad (Mx)_i \leq 1$ → d-dim P

$x_i = 0$ or $(Mx)_i = 1$ ⟶ Label/color i

- Vertex $v \in S_1 \setminus S$. Then $L(v) = \{2, \ldots, d\}$
  - Unique duplicate label



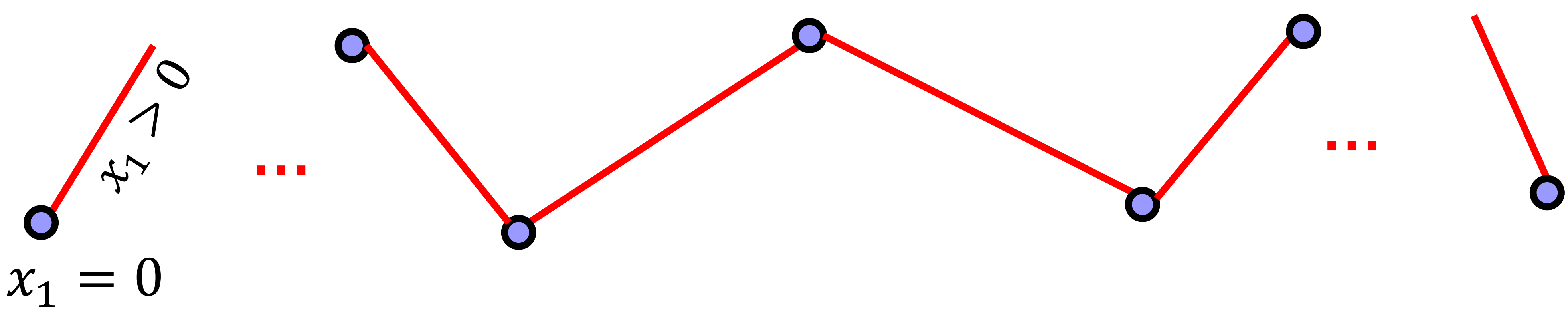
Cycle

S_1: Set of paths and cycles

$\forall i, x_i \geq 0, \quad (Mx)_i \leq 1 \rightarrow$ d-dim P

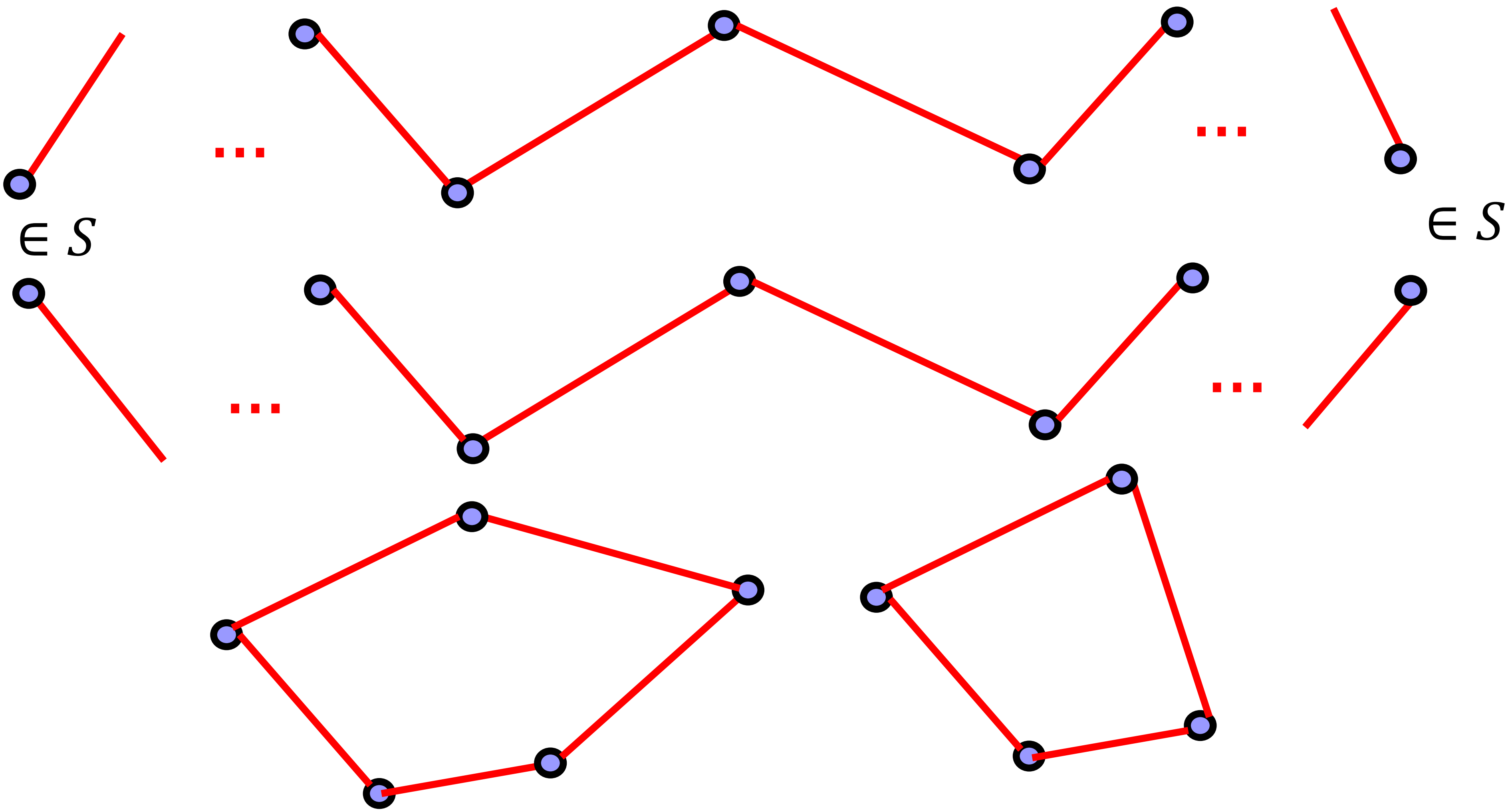
$x_i = 0$ or $(Mx)_i = 1 \longrightarrow$ Label/color i

- Vertex $v \in S(\subset S_1)$. Then $L(v) = \{1, \dots, d\}$
  - No duplicate label



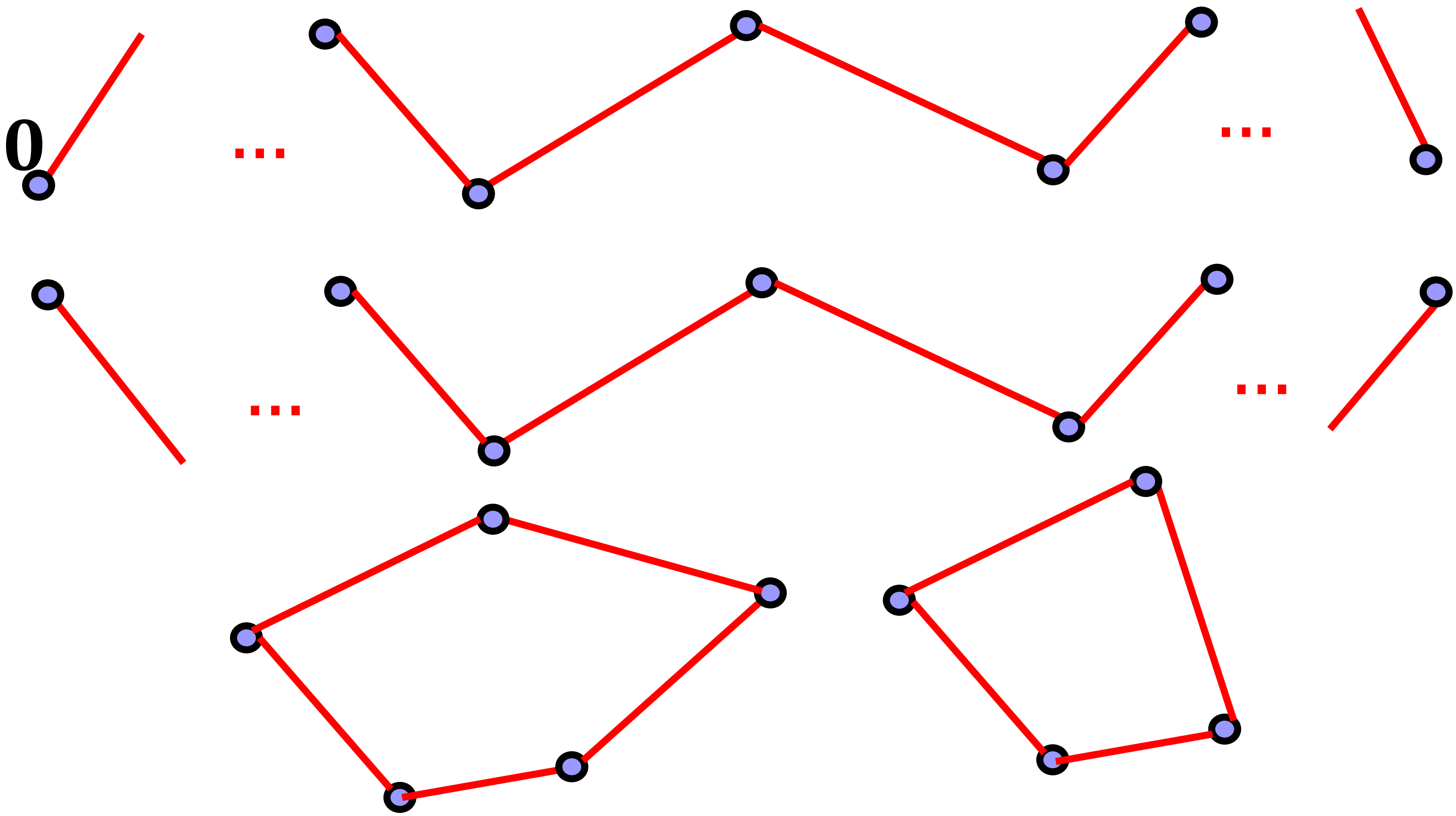
End in S

S_1: Set of paths and cycles



$S = \text{Solutions} \cup \{\mathbf{0}\}$

S_1: Set of paths and cycles

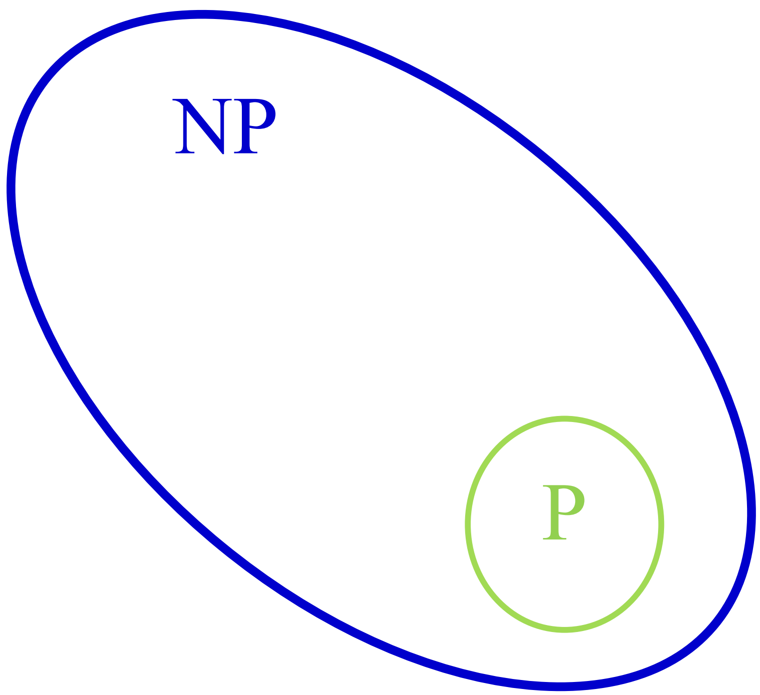


Goal: Find any other end-point   Defn of PPAD!

Computation? (in CS)

Not easy!

∃ solution?



What if solution always exists, like Nash Eq.?

Computation? (in CS)

Megiddo and Papadimitriou'91 :

Nash is NP-hard $\Rightarrow$ NP=Co-NP

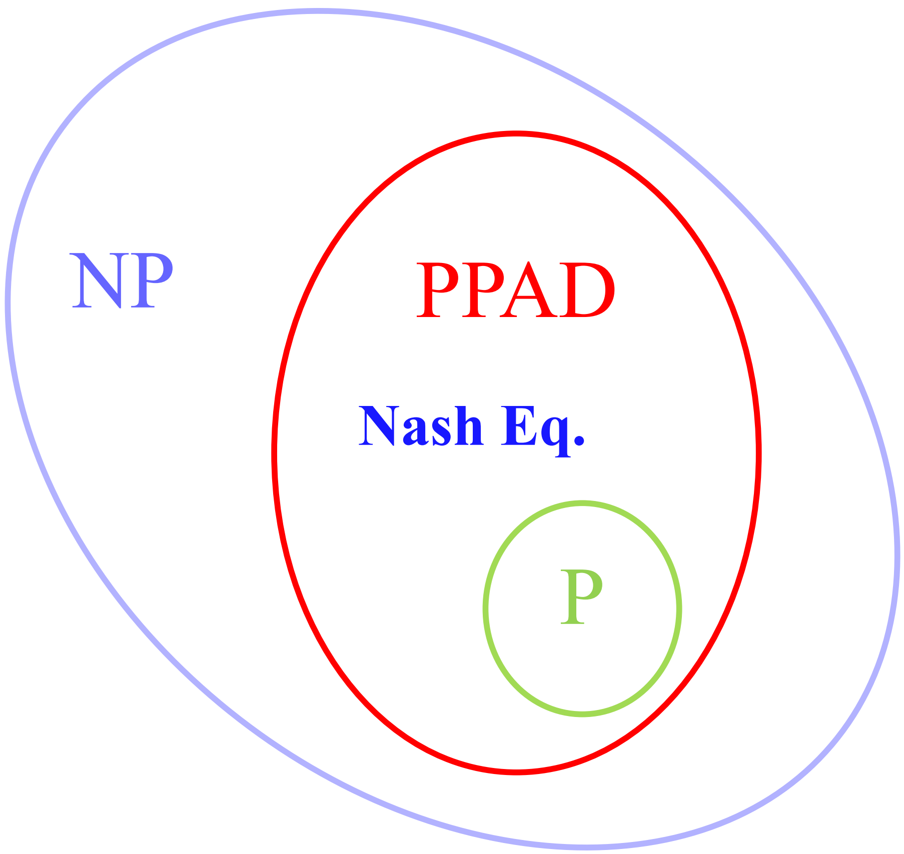
NP-hardness is ruled out!

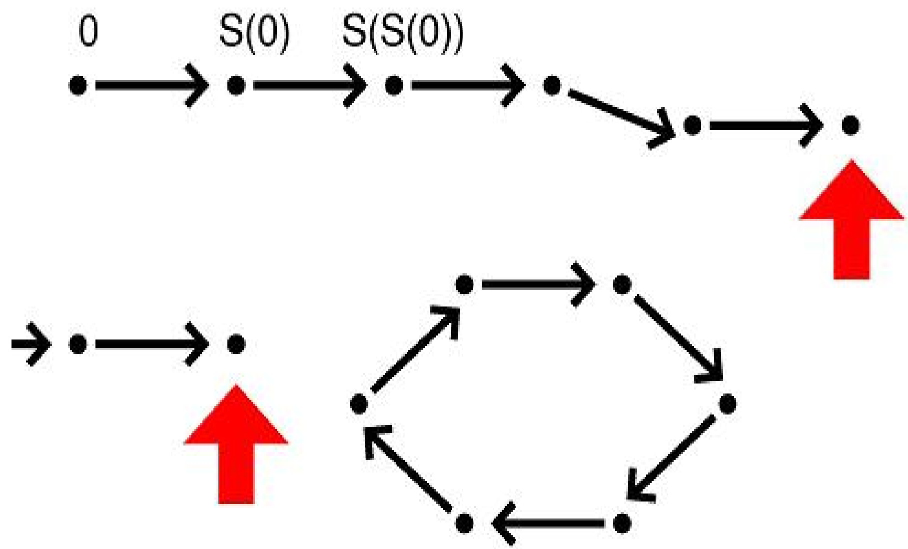
Complexity Classes

2-Nash is PPAD-complete!
[DGP'06, CDT'06]

Papadimitriou'94

PPAD   Polynomial Parity Argument for Directed graph





Find an end

Brute-force Algorithm?

P $\boxed{\begin{array}{c}\forall i, (Ay)_i \leq \pi_A \\ y \in \Delta_n\end{array}}$

Q $\boxed{\begin{array}{c}\forall j, (x^T B)_j \leq \pi_B \\ x \in \Delta_m\end{array}}$

Let (x, y) be a NE. Suppose we know supp(x) and supp(y).

Now can we find a NE?

Can we do better than “brute-force”?

Not so far. And may be never!

It is one of the hardest problems in PPAD.

What about special cases/approximation?

- Rank(A) or rank(B) is constant
- O(1)-approximate NE: quasi-polynomial time algorithm
- Constant rank games: rank(A+B) is a constant
  - FPTAS

P $\boxed{\begin{array}{c}\forall i, (Ay)_i \leq \pi_A \\ y \in \Delta_n\end{array}}$ Q $\boxed{\begin{array}{c}\forall j, (x^T B)_j \leq \pi_B \\ x \in \Delta_m\end{array}}$

$$(y, \pi_A, x, \pi_B) \in P \times Q$$

Theorem. If (A, B) is zero-sum, i.e., $A + B = 0$, then
2-Nash → linear programming

max: $-(\pi_A + \pi_B)$
s.t. $(y, \pi_A, x, \pi_B) \in P \times Q$

Rank of a game: rank(A+B)
Zero-sum ≡ Rank-0 games

$$P \quad \boxed{\begin{array}{c} \forall i, (Ay)_i \le \pi_A \\ y \in \Delta_n \end{array}} \qquad\qquad Q \quad \boxed{\begin{array}{c} \forall j, (x^T B)_j \le \pi_B \\ x \in \Delta_m \end{array}}$$

$$(y, \pi_A, x, \pi_B) \in P \times Q$$

Theorem. If (A, B) is zero-sum, i.e., $A + B = 0$, then

2-Nash → linear programming

Rank of a game: rank(A+B)

Poly-time approximation for constant rank games [KT'03].

Poly-time exact for rank-1 games [AGMS'11].

Exact for rank > 2 is PPAD-hard [M'13].

Open Problems

- Status of PPAD.
  - Is constant factor approximation of 2-Nash PPAD-hard?
- Not risk neutral? → Prospect Theory
  - Expected utility ≡ risk neutral