CS60025

ALGORITHMIC GAME THEORY

Game Theory:

Game is a system where a number of participants ~~is a~~ interact to reach an outcome. Every player derives some utility from the outcome.
The "outcome" depends on every player's action/strategy.

Our first example: Prisoners' Dilemma:

Two prisoners: P1, P2.

* Neither P1, nor P2 confesses: Both get 4 years of jail
* P1 confesses, P2 does not: P1 gets 1 year, P2 gets 8 years
* P2 confesses, P1 does not: P2 gets 1 year, P1 gets 8 years
* Both P1 & P2 confess: P1 & P2 get 2 years of jail.

The last option seems to be the best: called a stable solution

What if we consider the following:

* P1 and P2 neither confess: Both get 2 years
* Both confess: Both get 3 years
* If one confesses: They get 1 year, other gets 8 years.

Pay-off/utility matrix:

Player 1

Player 2	C	NC
C	(3, 3)	(1, 8)
NC	(8, 1)	(2, 2)

Normal Form Game

A normal form game is defined as:

$$\Gamma = (N, \{S_i\}_{i \in N}, \{u_i\}_{i \in N})$$

N : Set of players
S_i : Set of strategies (need not be finite)
u_i : utility function $\left[u_i : \underset{i \in N}{\times} S_i \to \mathbb{R}\right]$

For the prisoners dilemma,

$N = \{1, 2\}$
$S_1 = S_2 = \{C, NC\}$

1	2	u_1	u_2
C	C	3	3
C	NC	1	8
NC	C	8	1
NC	NC	2	2

The set $\underset{i \in N}{\times} S_i$ is called the set of <u>all strategic profiles</u>.

Assumptions:

* <u>Utility</u>: Every player has a utility function
* <u>Rationality</u>: Every player wants to maximize (or optimize) their utility.
* <u>Intelligence</u>: Every player has unbounded computational power.
* <u>Common Knowledge</u>: Any information J (e.g. Utility, rationality, intelligence), (every player knows that)* every player knows J.

Battle of the Sexes!

* Husband & Wife wants to watch a movie, husband wants to watch an action movie, ~~both~~ wife wants to watch a thriller (drama). They must agree on one movie.

		Player 2 (husband)	
		Drama	Action
Player 1 (wife)	Drama	(2, 1)	(0, 0)
	Action	(0, 0)	(1, 2)

Writing in the normal form:

$N = \{1, 2\}$

$S_1, S_2 = \{\text{Drama}, \text{Action}\}$

Coordination Game:

* Two companies want to manufacture 2 parts A & B. If both manufacture different parts, they gain nothing.

		Player 2	
		A	B
Player 1	A	(10, 10)	(0, 0)
	B	(0, 0)	(5, 5)

Rock, Paper, Scissors! — मेरा जाना उहिए!

Player 2

Player 1	R	P	S
R	(0, 0)	(-1, 1)	(1, -1)
P	(1, -1)	(0, 0)	(-1, 1)
S	(-1, 1)	(1, -1)	(0, 0)

win : 1
loss : -1.

- No stable ~~sets~~ solution

Matching Pennies / Coins

Player 1 wins if coins match, Player 2 wins o/w.

Player 2

Player 1	H	T
H	(1, -1)	(-1, 1)
T	(-1, 1)	(1, -1)

- No stable solutions

These (last two) are called "zero sum games" or "strictly competitive games"

- Exactly 2 players
- ∀ strategic profiles:
  utility of player 1 = - utility of player 2.

We will now see games having multiple players

Tragedy of the Commons:

- n players
- $S_i = \{0, 1\}$; $1 \equiv$ builds a factory, $0 \equiv$ does not build a factory.

  - 1 unit of utility for self
  - 5 units of harm shared between all players.

- $u_i(s_1, \ldots, s_i, \ldots, s_n) = s_i - \frac{5}{n}(s_1 + \cdots + s_n)$

Ideal situation : Nobody builds any factory, } Tragedy
But people do build factories }

Some Network-People-Bandwidth thingy

- n players
- Network with capacity 1.
- Player i uses bandwidth $x_i \in [0, 1]$
- Make sure $\sum_{i=1}^{n} x_i \leq 1$.
- Utility of i-th player : $x_i\left(1 - \sum_{j=1}^{n} x_j\right)$

$$= x_i\left(1 - \sum_{\substack{j=1 \\ i \neq j}}^{n} x_j - x_i\right)$$

$$= x_i(1 - t - x_i) \quad ; \; t = \sum_{\substack{j=1 \\ i \neq j}} x_j$$

maximized when $x_i = \frac{1-t}{2}$.

Best when $x_i = \frac{1}{n+1}$.

Bandwidth used is $\frac{n}{n+1}$, but individual pay-off is bit low

~~Art~~ Auctions:

- n seller and 1 buyer (?!? এরকম কি!!?).
- each having a valuation for an item.
- each bids a price.

$N = \{1, \dots, n\}$, $S_i = \mathbb{R}_{\geq 0}$, v_i = valuation of seller i, Allocation fn, $a: \prod_{i=1}^{n} S_i \to \mathbb{R}^N$

First price auction (smallest bid)	Second price auction (2nd smallest bid
Smallest bid wins	Second smallest bid But sold to the minimum bidder. কিভাবে হচ্ছে!

Allocation:

$$a : \prod_{i \in N} S_i \longrightarrow \{0,1\}^n$$

$$a\left((s_i)_{i \in N}\right) = (a_i)_{i \in N}, \quad a_i = \begin{cases} 1 & \text{if } s_i \leq s_j \ \forall j \in N \\ & s_i < s_j \ \forall j < i \\ 0 & \text{o/w} \end{cases}$$

(i.e. $a_i = 1$ iff i-th players bid chosen).

Payment fn: $p : \prod_{i \in N} S_i \to \mathbb{R}^N$, $p\left((s_i)_{i \in N}\right) = (p_i)_{i \in N}$

First price: $p_i = s_i a_i$
Second price: $p_i = \begin{cases} \min_{j \in N \setminus \{i\}} s_j & \text{if } a_i = 1 \\ 0 & \text{o/w} \end{cases}$

Utility:

$$u_i(s_1, s_2, \dots, s_n) = a_i(p_i - v_i)$$

Example:

s_i	100	80	50	90	120
v_i	*	*	30	*	*
p_i (FB)	*	*	20	*	*
p_i (SB)	*	*	50	*	*

Solution Concepts:

Dominant Strategy Equilibrium:

- **Strong Dominance:**

- A strategy $s_i^* \in S_i$ strongly dominates $s_i' \in S_i$ if $u_i(s_i^*, s_{-i}) > u_i(s_i', s_{-i})$, $\forall s_{-i} \in S_{-i}$, which is a tuple of strategy of other players. A strategy s_i^* is a strongly dominant strategy if s_i^* strongly dominates s_i', $\forall s_i' \in S_i \setminus \{s_i^*\}$

Example: prisoners' dillemma

P2 \ P1	C	NC
C	(-4,-4)	(-1,-8)
NC	(-8,-1)	(-2,-2)

C is a strongly dominant strategy for both prisoners.

- $(s_1^*, \dots, s_n^*)$ is a strongly dominant strategy equilibrium if $\forall i \in \{1, \dots, n\}$, s_i^* is a strongly dominant strategy for player i.

  Example: (C, C) is a strongly dominant strategy equilibrium, for the prisoners dilemma.

Weak Dominance:

Example: Variant of Prisoners Dilemma

	C	NC
C	(4, 4)	(2, 8)
NC	(8, 2)	(2, 2)

- $s_i' \in S_i$ weakly dominates $s_i \in S_i$ if $\forall s_{-i} \in S_{-i}$:

$$u_i(s_i', s_{-i}) \geq u_i(s_i, s_{-i})$$

and $\exists s_{-i} \in S_{-i}$ s.t. $u_i(s_i', s_{-i}) > u_i(s_i, s_{-i})$

- $s_i^* \in S_i$ is a weakly dominant strategy if s_i^* weakly dominates s_i, $\forall s_i \in S_i \setminus \{s_i^*\}$

- $(s_1^*, \dots, s_n^*)$ is a weakly dominant strategy equilibrium if s_i^* is a weakly dominant strategy for player i, $\forall i \in [n]$.

Example:

- "Battle of the sexes" have no weakly dominant strategy equilibrium.

- Consider the "Tragedy of the Commons" example:

n players: $u_i = s_i - \frac{5}{n}(s_1 + \cdots + s_n) = \frac{n-5}{n} s_i - \frac{5}{n} \sum_{j \neq i} s_j$

if $n < 5$: $(0, 0, \ldots, 0)$ is the "best" profile

if $n = 5$: (no strongly/weakly dominant strategy equilibrium)

if $n > 5$: $(1, 1, 1, \ldots, 1)$ is a dominant strategy profile.

- Consider that each factory owner has to pay 5 units of tax.

$$\therefore u_i(s_1, \ldots, s_n) = \left(\frac{n-5}{n}\right) s_i - \frac{5}{n} \sum_{j \neq i} s_j - (5 s_i)$$

$$= \left(\frac{-4n-5}{n}\right) s_i - \frac{5}{n} \sum_{j \neq i} s_j.$$

This time $(0, \ldots, 0)$ is the best profile

Very Weak Dominance:

Example: Consider another variant of the prisoner's dilemma.

	C	NC
C	(4, 4)	(2, 8)
NC	(4, 2)	(2, 2)

- $s_i \in S_i$ very weakly dominates $s_i' \in S_i$ if $u_i(s_i, s_{-i}) \geq u_i(s_i', s_{-i}) \quad \forall s_{-i} \in S_{-i}$.
- $s_i^* \in S_i$ is a very weakly dominant if s_i^* very weakly dominates s_i, $\forall s_i \in S_i$.
- $(s_1^*, \ldots, s_n^*)$ is a very weakly dominant equilibrium if s_i^* ~~do ve~~ is a very weakly dominant strategy, for every player $i \in [n]$.

Example: Second Price Auction.

$$a_i(s_1, \ldots, s_n) = \begin{cases} 1 & \text{if } s_i < s_j \ \forall j \in \{1, 2, \ldots, i-1\} \\ & \quad s_i \leq s_j \ \forall j > i \\ 0 & \text{otherwise.} \end{cases}$$

$$p_i(s_1, \ldots, s_n) = \begin{cases} \min_{j \neq i} s_j & \text{if } a_i(s_1, \ldots, s_n) = 1 \\ 0 & \text{otherwise.} \end{cases}$$

$$u_i(s_1, \ldots, s_n) = a_i(p_i - v_i)$$

Theorem: $(s_1, \ldots, s_n) = (v_1, \ldots, v_n)$ is a weakly dominant strategy equilibrium.

Proof: Let us just look at player 1. We consider the following two cases:

Case I: $v_1 \leq \min(s_2, \dots, s_n) = \theta$ (say).

if $s_1 > \min(s_2, \dots, s_n)$, seller 1 loses, $u_1 = 0 \leq \theta - v_1$
else if $s_1 \leq \min(s_2, \dots, s_n)$, seller 1 wins, $u_1 = \theta - v_1 \geq 0$.
So $s_1 = v_1$ gives you $u_1 = \theta - v_1 \geq 0$.

Case II: $v_1 > \min(s_2, \dots, s_n)$

if $s_1 \leq \theta$, seller 1 wins, $u_1 = \theta - v_1 < 0$
else $s_1 > \theta$, seller 1 loses, $u_1 = 0$
So $s_1 = v_1$, gives you $u_1 = 0 > \theta - v_1$

Exercise: The above proof just shows very weak dominance. We need to show that

$\exists\, s_2, \dots, s_n$. s.t.

$\forall\, s_1 \in \mathbb{R} \setminus \{v_1\}$

$u_1(v_1, s_2, \dots, s_n) > u_1(s_1, s_2, \dots, s_n)$.

NASH EQUILIBRIUM

Each player's strategy is the best possible among all players.

Pure strategy Nash equilibrium: (PSNE)

For $\Gamma = (N, (S_i)_{i \in N}, (u_i)_{i \in N})$, $(s_1^*, s_2^*, \dots, s_n^*)$ is a Pure Strategy Nash Equilibrium (PSNE) if

$\forall i \in [n]$, $u_i(s_i^*, s_{-i}^*) \geq u_i(s_i, s_{-i}^*) \quad \forall s_i \in S_i$

Best Response function: $B_i : S_{-i} \longrightarrow 2^{S_i}$

$B_i(s_{-i}) = \{s_i \in S_i : u_i(s_i, s_{-i}) \geq u_i(s_i', s_{-i}) \ \forall s_i' \in S_i\}$

Example: Battle of the Sexes:

	D	A
D	(2,1)	(0,0)
A	(0,0)	(1,2)

* No dominant strategy equilibrium
* (A,A) and (D,D) are nash equilibria.

Example: Prisoner's Dilemma

	C	NC
C	(-4,-4)	(-1,-8)
NC	(-8,-1)	(-2,-2)

* $B_1(C) = \{C\}$ $\quad B_1(NC) = \{C\}$
  $B_2(C) = \{C\}$ $\quad B_2(NC) = \{C\}$

(C, C) is a nash equilibrium.

Example: Tragedy of the commons

$$u_i(s_1, \dots, s_n) = s_i - \frac{5}{n} \sum_{j \neq i} s_j.$$

$n < 5$: $(0, 0, \dots, 0)$ is SDSE

$n > 5$: $(1, 1, \dots, 1)$ is SDSE

$n = 5$: $u_i(0, s_{-i}) = u_i(1, s_{-i}) \quad \forall i \in [n]$.

$B_i(s_{-i}) = \{0, 1\}$

No dominant strategy equilibrium, but all strategy profiles are PSNE.

Example: Channel Capacity Thingy

n players

x_i : flow of player-i

$$\sum_{i \in [n]} x_i < 1$$

payoff, $u_i = x_i \left(1 - \sum_{i \in [n]} x_i\right) = x_i (1 - t - x_i)$

We saw that $x_i^* = \frac{1}{n+1}$ gives the best possible

Individual best payoff $= \frac{1}{(n+1)^2}$

Collective payoff $= \frac{n}{(n+1)^2}$ —— "NOT GOOD"

* $\left(\frac{1}{n+1}, \frac{1}{n+1}, \ldots, \frac{1}{n+1}\right)$ is a PSNE.

* However the $\frac{n}{(n+1)^2}$ collective payoff is not that good, consider the profile

$$\left(\frac{1}{2n}, \frac{1}{2n}, \ldots, \frac{1}{2n}\right).$$

Individual payoff $= \frac{1}{2n}\left(1 - \frac{1}{2}\right) = \frac{1}{4n}$.

$\therefore$ Collective payoff $= \frac{1}{4} \gg \frac{n}{(n+1)^2}$.

- Nash Equilibrium need not be good collectively or individually.
- No PSNE in "matching pennies" or "rock-paper-scissors".

Mixed/Randomized Strategy Nash Equilibrium:

S_i be the set of pure strategies for the i-th player. Assume S_i's are finite. A mixed strategy σ_i for player i is a probability distribution over S_i. i.e. $\sigma_i : S_i \to [0,1]$ & $\sum_{i \in N} \sigma_i(s_i) = 1$

Assume $|S_i| = m$, and $\Delta(S_i)$ to be the set of all mixed strategies for player i.

$$\Delta(S_i) = \left\{ (\sigma_{i1}, \sigma_{i2}, \dots, \sigma_{im}) \in \mathbb{R}^m \;\middle|\; \sigma_{ij} \geq 0 \text{ and } \sum_{j \in [m]} \sigma_{ij} = 1 \right\}$$

Mixed Extension of Γ:

$$U_i : \underset{i \in N}{\times} \Delta(S_i) \to \mathbb{R}$$

$$U_i(\sigma_1, \dots, \sigma_n) = \sum_{(s_1, \dots, s_n) \in \underset{i \in N}{\times} S_i} \sigma(s_1, s_2, \dots, s_n)\, u_i(s_1, \dots, s_n)$$

where $\sigma(s_1, \dots, s_n) = \prod_{i \in [n]} \sigma_i(s_i)$

Randomness of independent players are mutually independent.

$$\therefore U_i(\sigma_1, \dots, \sigma_n) = \sum_{s_1 \in S_1} \sum_{s_2 \in S_2} \cdots \sum_{s_n \in S_n} \prod_{i \in [n]} \sigma_i(s_i) \cdot u_i(s_1, \dots, s_n).$$

Example: Battle of the Sexes:

σ_1, σ_2	D	A
D	(2,1)	(0,0)
A	(0,0)	(1,2)

$S_1 = S_2 = \{D, A\}$

$u_1(\sigma_1, \sigma_2) = 2\sigma_1(D)\sigma_2(D) + 1\,\sigma_1(A)\sigma_2(A)$.

$= 2\sigma_1(D)\sigma_2(D) + (1-\sigma_1(D))(1-\sigma_2(D))$

$= 1 - \sigma_1(D) - \sigma_2(D) + 3\sigma_1(D)\sigma_2(D).$

$u_2(\sigma_1, \sigma_2) = \sigma_1(D)\sigma_2(D) + 2\sigma_1(A)\sigma_2(A)$

$= \sigma_1(D)\sigma_2(D) + 2(1-\sigma_1(D))(1-\sigma_1(D))$

$= 2 - 2\sigma_1(D) - 2\sigma_2(D) + 3\sigma_1(D)\sigma_2(D).$

if $\sigma_1(D) = (2/3, 1/3)$ and $\sigma_2(D) = (1/3, 2/3)$.

$u_1 = u_2 = 2/3$ then.

Mixed Strategy Nash Equilibrium (MSNE):

$(\sigma_1^*, \ldots, \sigma_m^*)$ MSNE for Γ if $\forall i \in N$

$$u_i(\sigma_i^*, \sigma_{-i}^*) \geq u_i(\sigma_2, \sigma_{-i}^*) \quad \forall \sigma_i \in \Delta(S_i)$$

Best response function:

$$B_i(\sigma_{-i}) = \{\sigma_i \in \Delta(S_i) : u_i(\sigma_i, \sigma_{-i}) \geq u_i(\sigma_i', \sigma_{-i}) \quad \forall \sigma_i' \in \Delta(S_i)\}$$

$(\sigma_1^*, \ldots, \sigma_n^*)$ is MSNE iff $\sigma_i^* \in B_i(\sigma_{-i}^*)$ $\forall i \in N$.

Example: Battle of the Sexes:

$\sigma_1 = (2/3, 1/3)$ and $\sigma_2 = (1/3, 2/3)$, (σ_1, σ_2) is a MSNE.

$3\sigma_1^*(D)\sigma_2^*(D)$

This is because

$$u_1(\sigma_1^*, \sigma_2^*) \geq u_1(\sigma_1, \sigma_2^*) \quad \forall \sigma_1 \in \Delta(S_1)$$

i.e. $3\sigma_1^*(D)\sigma_2^*(D) - \sigma_1^*(D) \geq 3\sigma_1(D)\sigma_2^*(D) - \sigma_1(D)$

or, $\sigma_1^*(D)(3\sigma_2^*(D) - 1) \geq \sigma_1(D)(3\sigma_2^*(D) - 1)$

Case I: if $(3\sigma_2^*(D) - 1) > 0$

$\Rightarrow$ (D, D) is a PSNE (?)

$\Rightarrow$ (A, A) is a PSNE (?)

Case I: if $(3\sigma_2^*(D) - 1) > 0$

$\Rightarrow \sigma_1^*(D) \geq \sigma_1(D) \quad \forall \sigma_1 \in \Delta(S_1)$

$\Rightarrow \sigma_1^*(D) = 1. \rightsquigarrow \sigma_2^*(D) = 1$

Case II: if $(3\sigma_2^*(D) - 1) < 0$

$\Rightarrow \sigma_1^*(D) \leq \sigma_1(D) \quad \forall \sigma_1 \in \Delta(S_1)$

$\Rightarrow \sigma_1^*(D) = 0. \rightsquigarrow \sigma_2^*(D) = 0$

Case III if $(3\sigma_2^*(D) - 1) = 0.$

$\Rightarrow \sigma_1 = (2/3, 1/3), \sigma_2 = (1/3, 2/3).$

Exercise: Prove that the following are MSNE's

- uniform random in Matching Pennies
- uniform random in Rock-Paper-Scissors.

Support of σ_i, $\delta(\sigma_i) := \{s_i \mid \sigma_i(s_i) > 0\}$

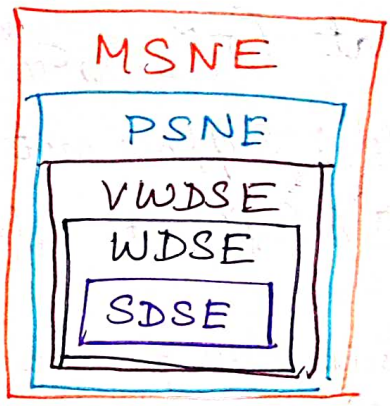
Lemma:

$(\sigma_1^*, \sigma_2^*, \sigma_3^*, \dots, \sigma_n^*)$ is a MSNE iff

(1) $u_i(s_i, \sigma_{-i}^*)$ is the same for every $s_i \in \delta(\sigma_i^*)$

(2) $u_i(s_i, \sigma_{-i}^*) \geq u_i(s_i', \sigma_{-i}^*)$, $\forall s_i \in \delta(\sigma_i^*)$, $\forall s_i' \notin \delta(\sigma_i^*)$

"Any finite game has an MSNE" — Nash's Theorem.



Proof of the Lemma:

[$\Leftrightarrow$] We would like to show:

$$u_i(\sigma_i^*, \sigma_{-i}^*) \geq u_i(\sigma_i, \sigma_{-i}^*), \quad \forall \sigma_i \in \Delta(S_i)$$

Look at $x_i = u_i(s_i, \sigma_{-i}^*)$ for $s_i \in \delta(\sigma_i^*)$ [by (1)]

Now,

$$u_i(\sigma_i^*, \sigma_{-i}^*) = \sum_{s_i \in S_i} \sigma_i^*(s_i)\, u_i(s_i, \sigma_{-i}^*)$$

$$= \sum_{s_i \in \delta(\sigma_i^*)} \sigma_i^*(s_i)\, u_i(s_i, \sigma_{-i}^*)$$

$$= x_i \sum_{s_i \in \delta(\sigma_i^*)} \sigma_i^*(s_i) = x_i.$$

$$\therefore u_i(\sigma_i^*, \sigma_{-i}^*) = x_i = x_i \sum_{s_i \in S_i} \sigma_i(s_i) \text{ for any } \sigma_i \in \Delta(S_i)$$

$$\overset{(2)}{\geq} u_i(s_i, \sigma_{-i}^*) \sum_{s_i \in S_i} \sigma_i(s_i)\, u_i(s_i, \sigma_{-i}^*)$$

$$= u_i(\sigma_i, \sigma_{-i}^*).$$

~~etc~~ (proved)

[⇒] Suppose that $(\sigma_1^*, \ldots, \sigma_n^*)$ is an MSNE.

Proof of (1) holds: Suppose not, i.e.

$\exists\ s_i, s_i' \in \delta(\sigma_i^*)$ s.t. $u_i(s_i, \sigma_{-i}^*) > u_i(s_i', \sigma_{-i}^*)$

~~However,~~

The contribution of these two terms is

~~$\sigma_i^*(s_i)(\sigma$~~

$$\sigma_i^*(s_i)\, u_i(s_i, \sigma_{-i}^*) + \sigma_i^*(s_i')\, u_i(s_i', \sigma_{-i}^*)$$

However, if we design a distribution σ_i' which is exactly the same as σ_i^*, but $\sigma_i'(s_i) = (\sigma_i^*(s_i) + \sigma_i^*(s_i'))$ and $\sigma_i'(s_i') = 0$

Therefore the ~~contradiction~~ contribution becomes

$$(\sigma_i^*(s_i) + \sigma_i^*(s_i'))\, u_i(s_i, \sigma_{-i}^*)$$

$$> \sigma_i^*(s_i)\, u_i(s_i, \sigma_{-i}^*) + \sigma_i^*(s_i')\, u_i(s_i', \sigma_{-i}^*)$$

, a contradiction!

(1) must hold then!

Proof that (2) holds: Suppose not, i.e.

$\exists\, s_i' \in \delta(\sigma_i^*)$ and $s_i'' \notin \delta(\sigma_i^*)$, s.t.

$$u_i(s_i'', \sigma_{-i}^*) > u_i(s_i', \sigma_{-i}^*).$$

Then we can interchange the probabilities of s_i' and s_i'' & get a strictly better contribution.

i.e.

$$u_i(\sigma_i^*, \sigma_{-i}^*) = \sum_{s_i \in S_i} \sigma_i^*(s_i) \cdot u_i(s_i, \sigma_{-i}^*)$$

$$= \sum_{s_i \in S_i \setminus \{s_i', s_i''\}} \sigma_i^*(s_i) \cdot u_i(s_i, \sigma_{-i}^*) + \sigma_i^*(s_i') u_i(s_i', \sigma_{-i}^*)$$

$$< \sum_{s_i \in S_i \setminus \{s_i', s_i''\}} \sigma_i^*(s_i) \cdot u_i(s_i, \sigma_{-i}^*) + \sigma_i^*(s_i') u_i(s_i'', \sigma_{-i}^*)$$

$$= u_i(\sigma_i', \sigma_{-i}^*),$$

where σ_i' is almost the same as σ_i^*, but probabilities of s_i' and s_i'' are interchanged.

$\Rightarrow \sigma_i'$ is "better" than σ_i^* ↯

This completes the proof, (hopefully)

Example: Battle of the Sexes, Revisited...

	D	A
D	(2,1)	(0,0)
A	(0,0)	(1,2)

$(1,0), (0,0)$; $(0,1), (0,1)$; $(\frac{2}{3},\frac{1}{3}), (\frac{1}{3},\frac{2}{3})$ are MSNE.

$\sigma_1 \quad \sigma_2$

There exist no NE with support.

$\{D\} \times \{D, A\}$ $\quad$ $\{D, A\} \times \{A\}$

$\{A\} \times \{D, A\}$ $\quad$ $\{D, A\} \times \{D\}$

$\{A\} \times \{D\}$ $\quad$ $\{D\} \times \{A\}$.

We can actually show that the $(\frac{2}{3},\frac{1}{3}), (\frac{1}{3},\frac{2}{3})$ is the only NE with support $\{D,A\} \times \{D, A\}$.

Proof: Suppose $\sigma_1^* = (x, 1-x)$ and $\sigma_2^* = (y, 1-y)$ is an MSNE. ~~$0 < x, y < 1$~~ with support $\{D, A\} \times \{D, A\}$, i.e. $0 < x, y < 1$.

Player 1:

$$u_1(D, \sigma_2^*) = u_1(A, \sigma_2^*)$$

$$\Rightarrow 2y = 1 \cdot (1-y) \Rightarrow y = 1/3.$$

Player 2:

$$u_2(D, \sigma_1^*) = u_2(A, \sigma_1^*)$$

$$\Rightarrow x = 2(1-x) \Rightarrow x = 2/3$$

Example: Coordination Game:

	C	M
C	(5,5)	(0,0)
M	(0,0)	(1,1)

~~Assume~~ (5,5), (1,1) are PSNE.

To find a MSNE with support $\{C,M\} \times \{C,M\}$

Let $\sigma_1^* = (x, 1-x)$ and $\sigma_2^* = (y, 1-y)$.

Player 1:

$$u_1(C, \sigma_2^*) = u_1(M, \sigma_2^*)$$

$$\Rightarrow 5 \cdot y = 1-y \Rightarrow y = 1/6$$

Player 2:

$$u_2(\sigma_1^*, C) = u_2(\sigma_2^*, M).$$

$$\Rightarrow x = 1/6.$$

Therefore $(1/6, 5/6)$, $(1/6, 5/6)$ is an MSNE.

Observe that they choose M with much more probability (2nd).

Matrix Games / 2 person - zero-sum games

Also called "strictly competitive games".
Can be described with an $m \times n$ matrix.

$$N = \{1, 2\}$$
$$S_1 = \{s_{11}, s_{12}, \dots, s_{1m}\}$$
$$S_2 = \{s_{11}, s_{22}, \dots, s_{2n}\}$$
$$u_1 = -u_2 .$$

$$A = (a_{ij})_{\substack{1 \le i \le n \\ 1 \le j \le m}} \qquad a_{ij} = u_1(s_{1i}, s_{2j}) = -u_2(s_{1i}, s_{2j})$$

Example : Matching Pennies

$$A = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

Example : Rock - Paper - Scissors

$$A = \begin{bmatrix} 0 & -1 & +1 \\ 1 & 0 & -1 \\ -1 & +1 & 0 \end{bmatrix}$$

Example : Product Prediction Game

Two companies, three products.

$$A = \begin{bmatrix} \boxed{1} & 2 & 1 \\ 0 & -1 & 2 \\ -1 & 0 & -2 \end{bmatrix}$$

→ PSNE.

$B_1(1) = \{1\}$ $B_2(1) = \{1, 3\}$
$B_1(2) = \{1\}$ $B_2(2) = \{2\}$
$B_1(3) = \{2\}$ $B_2(3) = \{3\}$

$1 \in B_1(1)$ and $1 \in B_2(1)$

[Recall that PSNE (i, j) $\equiv$ $i \in B_2(j)$ and $j \in B_1(i)$

$(1,1)$ happens to be a minimum in its row & a maximum in its colourmn. — This is called a saddle point; all saddle points are PSNE's

Saddle Points.

Def

$A = [a_{ij}]$: $m \times n$ matrix,

a_{ij} is a saddle point of A, if

$a_{ij} \leq a_{il}$, $\forall l \in [n]$

$a_{ij} \geq a_{kj}$, $\forall k \in [n]$

Def

If a_{ij} is a saddle point, then $a_{ij} = u_1(s_{1i}, s_{2j})$ is called the value of the game

Proposition 08.08.2024-1: a_{ij} is a saddle point iff (s_i, s_j) is a PSNE.

Proof: a_{ij} is a saddle point

$\Leftrightarrow$ a_{ij} is row max in A, a_{ij} is col max in A

$\Leftrightarrow$ $-a_{ij}$ is row max in $-A$, a_{ij} is col max in A

$\Leftrightarrow$ s_{1i} is a best response of row player against s_{2j} & s_{2j} is a best response of row player against s_{1i}

$\Leftrightarrow$ (s_{1i}, s_{2j}) is PSNE (definition of PSNE).

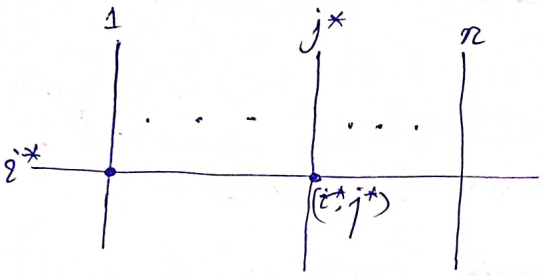
Theorem 08.08.2024-2: Let

$$u_R = \max_i \underbrace{\min_j a_{ij}}_{\text{maximum of row mins}} \quad \text{and} \quad u_C = \min_j \underbrace{\max_i a_{ij}}_{\text{minimum of col maxs.}}$$

A has a saddle point iff $u_R = u_C$, $= \text{val. of game}$

Proof:

[$\Rightarrow$] Let A have the saddle point $a_{i^*j^*}$



$$\min_j a_{i^*j} = a_{i^*j^*}$$

$$\max_i a_{ij^*} = a_{i^*j^*}$$

~~$a_{i^*j^*}$ is the col max of $j^* \Rightarrow$ min max~~

moreover, $\forall j$, $\max_i a_{ij} \geq a_{i^*j}$

$$\Rightarrow \min_j \max_i a_{ij} \geq \min_j a_{i^*j} = a_{i^*j^*}$$

Now, $\max_i a_{ij^*} = a_{i^*j^*}$

$$\Rightarrow \min_j \left(\max_i a_{ij^*} \right) \leq \min_j \max_i a_{ij} = a_{i^*j^*}$$

$$\Rightarrow \min_j \max_i a_{ij} = a_{i^*j^*}$$

$$\Rightarrow u_C = a_{i^*j^*}$$

An identical symmetric argument shows

$$u_R = a_{i^*j^*} \Rightarrow u_R = u_C.$$

[$\Leftarrow$] Exercise

- All saddle points have the same value.
- Value of game is unique
- If a_{ij} and a_{hk} are saddle points, so are a_{ik} & a_{hj}.

Example: $A = \begin{bmatrix} 5 & \textcircled{3} & 5 & \textcircled{3} \\ 2 & 1 & -1 & -2 \\ 4 & \textcircled{3} & 5 & \textcircled{3} \end{bmatrix}$

Looking into ~~PSNE~~ MSNE's:

$x = (x_1, \ldots, x_m)$ be prob. dist. of player 1.
$y = (y_1, \ldots, y_n)$ be prob. dist. of player 2.

$$u_1(x,y) = \sum_{i=1}^{m} \sum_{j=1}^{n} x_i y_j u_1(s_{1i}, s_{2j})$$

$$= \sum_{i=1}^{m} \sum_{j=1}^{n} x_i y_j a_{ij} = xAy^T$$

$$u_2(x,y) = -xAy^T$$

Assured payoff for player 1 on playing $x = \min_{y \in \Delta(S_2)} (xAy^T)$

Assured payoff for player 2 on playing $y = \min_{x \in \Delta(S_1)} (-xAy^T)$

loss of player 2 $= \max_{x \in \Delta(S_1)} xAy^T$

Goal of player 1: maximize payoff

$$\max_{x \in \Delta(S_1)} \min_{y \in \Delta(S_2)} xAy^T$$

Goal of player 2: minimize loss

$$\min_{y \in \Delta(S_2)} \max_{x \in \Delta(S_1)} xAy^T$$

Lemma 08.08.2024-3:

① When row player plays x, then among the effective strategies y, of the column player, $\exists$ at least one pure strategy.

i.e. $\min_{y \in \Delta(S_2)} x A y^T = \min_j \sum_{i=1}^{m} a_{ij} x_i$

② When col. player plays y, then among the effective strategies x of the row player, $\exists$ atleast one pure strategy.

i.e. $\max_{x \in \Delta(S_1)} x A y^T = \max_i \sum_{j=1}^{n} a_{ij} y_j$

Proof of (1), ((2) can be proved similarly):

Of course $\underbrace{\min_j \sum_{i=1}^{m} a_{ij} x_i}_{\text{a pure strategy i.e. a special mixed strategy (assign prob 1 to a minimum, & 0 to all).}} \geq \underbrace{\min_{y \in \Delta(S_2)} x A y^T}_{\text{minimum over all mixed strategies.}}$

$$x A y^T = \sum_{j=1}^{n} y_j \left(\sum_{i=1}^{m} a_{ij} x_i \right)$$

$$\geq \sum_{j=1}^{n} y_j \left(\min_j \sum_{i=1}^{m} a_{ij} x_i \right)$$

$$= \left(\min_j \sum_{i=1}^{m} a_{ij} x_i \right) \sum_{j=1}^{n} y_j = \min_j \sum_{i=1}^{m} a_{ij} x_i$$

$\therefore \forall x \in \Delta(S_1), y \in \Delta(S_2)$,

$$x A y^T \geq \min_j \sum_{i=1}^{m} a_{ij} x_i$$

$$\Rightarrow \min_{y \in \Delta(S_2)} x A y^T \geq \min_j \sum_{i=1}^{m} a_{ij} x_i \;, \quad \forall x \in \Delta(S_1)$$

$$\Rightarrow \min_{y \in \Delta(S_2)} x A y^T = \min_j \sum_{i=1}^{m} a_{ij} x_i$$

Restating the Optimization Problem

ROW PLAYER (P1)

$$\text{maximize} \quad \min_j \sum_{i=1}^{m} a_{ij} x_i$$

$$\text{subject to} \quad \sum_{i=1}^{m} x_i = 1$$

$$x_i \geq 0 \quad \forall i \in [m]$$

COLUMN PLAYER (P2)

$$\text{minimize} \quad \max_i \sum_{j=1}^{n} a_{ij} y_j$$

$$\text{subject to} \quad \sum_{j=1}^{n} y_j = 1$$

$$y_j \geq 0 \quad \forall j \in [n]$$

These can be formalised as linear programs.

LP 1

$$\text{maximize} \quad z$$

$$\text{subject to} \quad z - \sum_{i=1}^{m} a_{ij} x_i \leq 0 \quad \forall j \in [n]$$

$$\sum_{i=1}^{m} x_i = 1$$

$$x_i \geq 0 \quad \forall i \in [m]$$

LP2

$$\text{minimize} \quad w$$

$$\text{subject to} \quad w - \sum_{j=1}^{n} a_{ij} y_j, \quad \forall i \in [m]$$

$$\sum_{j=1}^{n} y_j = 1$$

$$y_j \geq 0 \quad \forall j \in [n]$$

Claim 08.08.2024 – 4: P1 & LP1 are equivalent.

Proof: Let $z^*, x_1^*, \ldots, x_m^*$ is an optimal solution to LP1.

Let j^* be an index s.t. $j^* \in [n]$ and

$$\min_j \sum_{i=1}^{m} a_{ij} x_i^* = \sum_{i=1}^{m} a_{ij^*} x_i^*$$

$$\Rightarrow \forall j \in [n] \quad \sum_{i=1}^{m} a_{ij^*} x_i^* \leq \sum_{i=1}^{m} a_{ij} x_i^*$$

Notice that $z^* = \min_j \sum_{i=1}^{m} a_{ij} x_i^*$ must be the case. Otherwise if ~~$z^* \Rightarrow m$~~ $z^* < \sum_{i=1}^{m} a_{ij^*} x_i^*$, we can pick $z' = \sum_{i=1}^{m} a_{ij^*}$, such that $z', x_1^*, \ldots, x_m^*$ is a better solution to LP1 than $z^*, x_1^*, \ldots, x_m^*$; contradiction –

$\therefore$ maximizing z,

$\Rightarrow$ maximizing $\min_j \sum_{i=1}^{m} a_{ij} x_i$.

$\therefore P_1 \equiv LP1$. ▨

(Same with P2 ≡ LP2

General form of a linear program :

Primal LP:

maximize cx^T
subject to $Ax \leq b$
$x \geq 0$

Dual LP:

minimize wb^T
subject to $wA \geq c$
$w \geq 0$.

Strong Duality Theorem: If both the primal and dual are feasible, then they have optimal solutions, and the values corresponding to the optimal solutions are equal; i.e if x^* is an optimal solution for primal L.P. & if y^* is an optimal solution for the dual L.P., then,

$$cx^{*T} = w^* b^T$$

Theorem 09.08.2024-1 (Minimax Theorem): For every $m \times n$ matrix A, $\exists$ stochastic vectors x^*, y^* s.t.

$$\min_{y \in \Delta(S_2)} (x^* A y^T) = \max_{x \in \Delta(S_1)} (x A y^{*T}).$$

Proof: Clearly LP1 and LP2 are feasible. ~~Therefore by strong duality theorem, if~~ $\exists z^*, x_1^*, \ldots, x_m^*$ is an optimal solution for LP1, and

Let $j^* \in [n]$, s.t.

$$z^* = \sum_{i=1}^{m} x_i^* a_{ij^*}.$$

$$\therefore \sum_{i=1}^{m} x_i^* a_{ij^*} = z^* \leq \sum_{i=1}^{m} x_i^* a_{ij} \quad \forall j \in [n].$$

$$\therefore \sum_{i=1}^{m} a_{ij^*} x_i^* = \min_j \sum_{i=1}^{m} a_{ij} x_i^*$$

$$= \min_{y \in \Delta(S_2)} x^* A y^T \quad \text{(by Lemma 08-08-2024-3)}$$

$\therefore$ Value of LP1 $= z^* = \min_{y \in \Delta(S_2)} x^* A y^T$.

Similarly if $w^*, y_1^*, \ldots, y_n^*$ is an optimal solution of LP2, then

$$w^* = \sum_{j=1}^{n} a_{i^*j} y_j^* \quad \text{, for some } i^* \in [m]$$

and ~~$w^* = \max$~~

$$w^* = \sum_{j=1}^{n} a_{i^*j} y_j^* = \max_i \sum_{j=1}^{n} a_{ij} y_j^*$$

$$= \max_{x \in \Delta(S_1)} x A (y^*)^T \quad \text{(by Lemma 08-08-2024-3)}$$

Now, by strong duality theorem,

value of primal LP (LP1) = value of dual LP (LP2)

$\therefore z^* = w^*$

$\Rightarrow \min_{y \in \Delta(S_2)} x^* A y^T = \max_{x \in \Delta(S_1)} x A (y^*)^T$.

□

Note: There exists polytime algorithm (Ellipsoid Algorithm). There we can get x^* and y^* in polytime.

Claim 09.08.2024-2: x^* & y^* be optimal solutions for LP1 & LP2 respectively, then (x^*, y^*) is an MSNE

Proof:

$$x^* A (y^*)^T \geq \min_{y \in \Delta(S_2)} x^* A y^T$$

$$= \max_{x \in \Delta(S_1)} x A (y^*)^T \quad , [\text{from Minimax Theorem}]$$

$$\geq x A (y^*)^T \quad \forall x \in \Delta(S_1).$$

$\therefore$ x^* is a best strategy against y^*.

Again,

$$x^* A (y^*)^T \leq \max_{x \in \Delta(S_1)} x A (y^*)^T$$

$$= \min_{y \in \Delta(S_2)} x^* A y^T \quad , [\text{from Minimax Theorem}]$$

$$\leq x^* A y^T \quad \forall y \in \Delta(S_2).$$

$\therefore$ y^* is a best strategy against x^*.

$\therefore$ $u_1(x^*, y^*) \geq u_1(x, y^*) \quad \forall x \in \Delta(S_1)$

and, $-u_2(x^*, y^*) \geq -u_2(x^*, y) \quad \forall y \in \Delta(S_2)$

$\Rightarrow (x^*, y^*)$ is an MSNE. $\square$

Example: Rock-Paper-Scissors:

$$A = \begin{bmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{bmatrix}$$

maximize z

subject to

$z - (x_2 - x_3) \leq 0$

$z - (-x_1 + x_3) \leq 0$

$z - (x_1 - x_2) \leq 0$

$x_1 + x_2 + x_3 = 1$

$x_1, x_2, x_3 \geq 0$.

$$z^* = 0$$
$$x_1^* = x_2^* = x_3^* = 1/3$$

Max-Min Inequality

- $\max_{x \in \Delta(S_1)} \min_{y \in \Delta(S_2)} x A y^T \leq \min_{y \in \Delta(S_2)} \max_{x \in \Delta(S_1)} x A y^T$

- In the general setting,

  $\forall\ f : X \times Y \longrightarrow \mathbb{R}$, $\forall X, Y$

  $$\sup_{x \in X} \inf_{y \in Y} f(x,y) \leq \inf_{y \in Y} \sup_{x \in X} f(x,y).$$

- If $X, Y \subset \mathbb{R}^n$ and X, Y are compact and convex, and $f(\cdot, y)$ ~~$f(x,\cdot)$~~ is concave, and $f(x,\cdot)$ is convex, then equality holds.

(This holds true for functions like xAy^T)

(We have already proved something stronger).

Proof: ~~max~~ R.H.S.

$$= \min_{y \in \Delta(S_2)} \max_{x \in \Delta(S_1)} x A y^T$$

$$= \min_{y \in \Delta(S_2)} \max_{i \in [m]} \sum_{j=1}^{n} a_{ij} y_j$$

$$= \min_{y \in \Delta(S_2)} \left\{ \left(\sum_{i=1}^{m} x_i^* \right) \left(\max_{i \in [m]} \sum_{j=1}^{n} a_{ij} y_j \right) \right\}, \quad [\text{for any dist. } x_i^*]$$

$$\geq \min_{y \in \Delta(S_2)} \left\{ \sum_{i=1}^{m} x_i^* \sum_{j=1}^{n} a_{ij} y_j \right\} = \min_{y \in \Delta(S_2)} x^* A y^T$$

We choose x^* s.t. $\left(\min_{y \in \Delta(S_2)} x A y^T\right)$ is maximized.

$\therefore$ L.H.S

$$= \max_{x \in \Delta(S_1)} \min_{y \in \Delta(S_2)} x A y^T$$

$$= \min_{y \in \Delta(S_2)} x^* A y^T \leq \text{R.H.S.} \qquad \square.$$

Yao's Lemma:

Used to prove lower bounds on worst-case performance of randomized algorithms.

Let Π be a problem, $\mathcal{X}$ be the set of all inputs to Π, $\mathcal{A}$ be the set of all deterministic algorithms for Π.

Let $A(x; r)$ be a randomized algorithm upon input x and randomness r. Therefore $A(x; r)$ is a distribution over $\{A(x; r)\}_{r \in R}$, which is a set of deterministic algorithms.

Let $T(A, x)$ be cost of algorithm A on input x.

Lemma 16.08.2024-1 (Yao's Lemma):

$A \xleftarrow{\sigma_A} \mathcal{A} \qquad X \xleftarrow{\sigma_X} \mathcal{X}$

$$\max_{x \in \mathcal{X}} \mathbb{E}[T(A, x)] \geq \min_{a \in \mathcal{A}} \mathbb{E}[T(a, X)]$$

worst case runtime of any randomized algorithm $\geq$ Runtime of best deterministic algorithm against random input $X \xleftarrow{\sigma_X} \mathcal{X}$

"For any randomized algorithm A, ∃ a (∀?) distribution X s.t. expected cost of A on its worst input is at least the cost of the best deterministic algorithm on the random input according to X."

Proof of Yao's Lemma:

B be a matrix game with rows indexed by $\mathcal{X}$ and columns indexed by $\mathcal{A}$.

∴ By max-min inequality,

$$\min_{z \in \Delta(\mathcal{A})} \max_{w \in \Delta(\mathcal{X})} w B z^T \geq \max_{w \in \Delta(\mathcal{X})} \min_{z \in \Delta(\mathcal{A})} w B z^T$$

Let e_x be the thing that assigns 1 to x, 0 to everything else (a pure strategy)

$$\therefore \max_{w \in \Delta(\mathcal{X})} w B z^T = \max_{x \in \mathcal{X}} e_x B z^T$$

and $$\min_{z \in \Delta(\mathcal{A})} w B z^T = \min_{a \in \mathcal{A}} w B e_a^T.$$

∴ Max-min inequality becomes:

$$\min_{z \in \Delta(\mathcal{A})} \max_{x \in \mathcal{X}} e_x B z^T \geq \max_{w \in \Delta(\mathcal{X})} \min_{a \in \mathcal{A}} w B e_a^T.$$

Now,

$$\max_{x \in \mathcal{X}} e_x B \sigma_A^T \geq \min_{z \in \Delta(\mathcal{A})} \max_{x \in \mathcal{X}} e_x B z^T.$$

$$\geq \max_{w \in A(x)} \min_{a \in \mathcal{A}} w B e_a^T \quad \text{[From max-min ineq.]}$$

$$\geq \min_{a \in \mathcal{A}} \sigma_x B e_a^T. \quad \text{---------- (i)}$$

Now, we define $B = (b_{x,a})_{\substack{x \in \mathcal{X} \\ a \in \mathcal{A}}}$ as.

$$b_{x,a} = T(a,x) \quad \forall x \in \mathcal{X}, a \in \mathcal{A}.$$

Now, $e_x B \sigma_A^T = \mathbb{E}[T(A,x)]$

and $\sigma_X B e_a^T = \mathbb{E}[T(a,X)]$

$$\therefore \max_{x \in \mathcal{X}} \mathbb{E}[T(A,x)] \geq \min_{a \in \mathcal{A}} \mathbb{E}[T(a,X)] \quad \square$$

Randomized comparison based sorting algorithms:

Theorem 16.08.2024-2: Randomized comparison based sorting algorithms have expected runtime (no. of comparison) of $\Omega(n \log n)$.

Proof: $\mathcal{A}$: Set of all deterministic sorting algorithm.
$\mathcal{X}$: Set of all possible inputs.

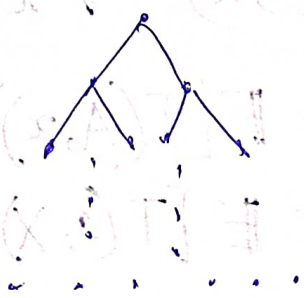
Comparison based sorting algorithms can be viewed as decision trees, where its height is the runtime.

We wish to show that $\max_{x \in \mathcal{X}} \mathbb{E}[T(A,x)] = \Omega(n \log n)$

By Yao's Lemma, it is sufficient to show one distribution ~~for~~ X for which

$$\min_{a \in A} E[T(a, X)] = \Omega(n \log n).$$

Take X to be the uniform distribution.



$a \in A$

leaf nodes would be one of $n!$ permutations. Therefore any correct algorithm must have $\geq n!$ leaf nodes.

Height of any ~~tree~~ such tree $= \log n!$

$$= \Omega(n \log n)$$

This is the best possible average height. (?)

$$E[T(a, X)] = \frac{1}{n!} \sum_{x \in X} T(a, x) = \Omega(n \log n).$$

If the tree is balanced average height is clearly $\Omega(n \log n)$. If it is not balanced, take two ~~dead~~ leaves of max height h_{max}. Remove them, make them the children of the shallowest leaf. The average

height decreases, and we can continue doing this till the tree becomes balanced.

Δ avg-ht $= -2h_{max} - h_{min} + 2(h_{min}+1) + h_{max} - 1 \leq -h_{max} + h_{min} + 1$.

when $h_{max} - h_{min} \geq 2$, Δ avg-ht < 0.

$\therefore$ $\log n!$ is indeed the min possible average height. □

Computing Equilibria:

$\Gamma = (N, (S_i)_{i \in N}, (u_i)_{i \in N})$ be a game,

$\sigma : (\sigma_1, \sigma_2, \ldots, \sigma_n)$ be a mixed strategy profile.

Support of σ_i : $\delta(\sigma_i) = \{s \in S_i : \sigma_i(s) > 0\}$

Support of σ : $\delta(\sigma) := \delta(\sigma_1) \times \delta(\sigma_2) \times \cdots \times \delta(\sigma_n)$.

"If the strategy sets are finite (S_i's are finite), even though the number of strategic profiles are finite, the number of subsets of $S_1 \times S_2 \times \cdots \times S_n$ that form supports of Nash Equibria is finite."

Example: In the 'battle of the sexes', $\{D\} \times \{D\}$, $\{A\} \times \{A\}$ and $\{DA\} \times \{DA\}$ were the only supports having Nash Equilibria.

$\#\text{supports} = (2^{|S_1|} - 1)(2^{|S_2|} - 1) \cdots (2^{|S_n|} - 1)$

Support Enumeration Method: Check if on all these supports which have a Nash equilibrium.

Recall the necessary & sufficient condition for MSNE:

$(\sigma_1, \ldots, \sigma_n)$ is an MSNE iff

- $u_i(s_i, s_{-i})$ is the same $\forall s_i \in \delta(\sigma_i)$
- $u_i(s_i, s_{-i}) \geq u_i(s_i', s_{-i})$ $\forall s_i \in \delta(\sigma_i)$, $\forall s_i' \notin \delta(\sigma_i)$

~~Run through of an Example~~

Let guess of support be ~~$W_1 \times W_2 \times \cdots \times W_n$~~ $X_1 \times X_2 \times \cdots \times X_n$.

NE exists $\Rightarrow \exists\, w_1, w_2 \ldots w_n$ and mixed strategies $\sigma_1, \ldots, \sigma_n$ s.t.

(1) $w_i = \sum_{s_{-i} \in S_{-i}} \left(\prod_{j \neq i} \sigma_j(s_j) \right) u_i(s_i, s_{-i})$ $\forall s_i \in X_i$, $\forall i \in N$

(2) $w_i \geq \sum_{s_{-i} \in S_{-i}} \left(\prod_{j \neq i} \sigma_j(s_j) \right) u_i(s_i, s_{-i})$ $\forall s_i \in S_i \setminus X_i$

(3) $\sigma_i(s_i) > 0$ $\forall s_i \in X_i$

(4) $\sigma_i(s_i) = 0$ $\forall s_i \in S \setminus X_i$

(5) $\sum_{s_i \in S_i} \sigma_i(s_i) = 1$.

Number of (1) equations : $|X_1| + |X_2| + \cdots + |X_n|$
Number of (2) equation : $|S_1 \setminus X_1| + \cdots + |S_n \setminus X_n|$
Number of (3) equations : $|X_1| + \cdots + |X_n|$
Number of (4) equations : $|S_1 \setminus X_1| + |S_2 \setminus X_2| \cdots$
Number of (5) equations : n

Total : $n + \underbrace{2 \sum_{i=1}^{n} |S_i|}_{\text{too large}}$.

- For bimetric games, this becomes linear and can be solved with techniques to solve "linear complementarity problem. (Which is hard)
- "For more players, things become non-linear; making it much more hard."

"In most cases we don't even know if we can write down MSNE's in finite space (remember that things can be irrational!"

Representation of Two Player Games (and beyond)

(a). $\Gamma_1 = \{\{1,2\}, (S_1, S_2), (A_1, A_2)\}$

$|S_1| = m \quad |S_2| = n$

A_1 : $m \times n$ matrix
A_2 : $m \times n$ matrix

Two player

$\therefore$ $2mn$ numbers to ~~representation~~

(b) Multiple Players.

$$\Gamma_2 = \{[n], (S_i)_{i\in N}, (u_i)_{i\in N}\}$$

$$|S_i| = s \quad \forall i \in N$$

ns^n are required to represent Γ_2.

$$\#\text{samples} = (2^s)^n = \text{poly}(ns^n) \text{ if } s \text{ is constant.}$$

However this does not mean the algorithm is polytime.

Succint Games:

Graphical Games:

There is a graph (directed) $(\{1, \ldots, n\}, E)$

s.t. $(i,j) \in E$ if the utility of player j depends on strategy chosen by player i, $(i \neq j)$, (of course j's strategy affects j's utility).

Graphical games with indegree $\leq d$.

$\therefore$ $n \cdot s^{d+1}$ numbers required to represent this.

For a small value of d, this is much better/smaller than $n \cdot s^n$.

Sparse Games : Very few of ns^n values are non-zero.

(Only non-zero values represented)

Exercise : Think of such games where

- n = large, d is small (1 and 2)
- $s = 2$.

Symmetric Games :

All players are identical. Utility of a player in a strategy profile depends on how many players play each of the s strategies.

For two player games this means

$$A_1 = (A_2)^T.$$

For this, only the count of strategies matter. If x_1 players play s_1, x_2 players play s_2, ... then we have $\binom{n+s-1}{s-1}$ allocations. Further we need to fix which strategy the i-th player played.

$$s \cdot \binom{n+s-1}{s-1} \text{ numbers are required.}$$

Multi-matrix games:

n players, $|S_i| = m \quad \forall i \in N$.

$\forall (i,j) \in N \times N$, $m \times m$ utility matrix given by A^{ij}.

Utility of player i in strategy profile $(s_i)_{i \in N}$ is

$$\sum_{i \neq j} A^{ij}_{s_i s_j}$$

$A^{ij}_{s_i s_j}$: entry at column s_j row s_i of matrix A^{ij}.

$m^2(n^2 - n)$ numbers required

~~Congestion Games~~

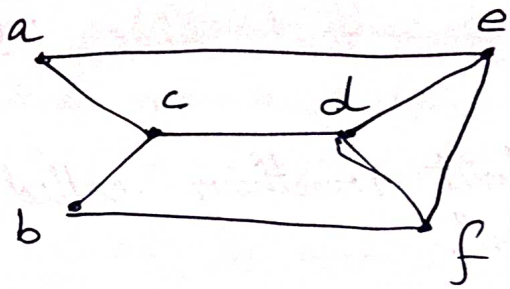
Network Congestion Games:

- Graph G with edge set E. Every player i has a source vertex & a destination vertex (s_i, t_i).
- Strategies of player i: Paths from s_i to t_i.

$P = (P_i)_{i \in N}$ be a strategy profile.

load $l_e(P) = \#$paths in $(P_i)_{i \in N}$ going over e.

Example:



player	(s_i, t_i)	P_i
1	(a, e)	a, c, d, e
2	(b, e)	b, c, d, e

$\ell_{cd}(P) = 2$.

load on cd.

$$\ell_{cd}(P) = |\{i \in N \mid e \in P_i\}|$$

Congestion function $c_e : \{0, 1, \ldots, n\} \longrightarrow \mathbb{N} \cup \{0\}$.

utility $u_i(P) = -\sum_{e \in P_i} c_e(\ell_e(P))$

Goal of players is to minimize costs.

"These sort of games will have ~~more~~ Nash Equilibria".

The questions we asked:

(i) Does there exist NE?
(ii) If yes, can we compute it.
(iii) If yes, how fast?

Potential Games:

A class of games that have PSNE

These games admit something called a "potential function".

$\Gamma = (N, (S_i)_{i \in N}, (u_i)_{i \in N})$ is called a potential game, if $\exists$ a function

$\phi : S_1 \times S_2 \times \cdots \times S_n \longrightarrow \mathbb{R}$ such that

$\forall i \in N, \forall s_i, s_i' \in S_i, \forall s_{-i} \in S_{-i}$

$$\phi(s_i, s_{-i}) - \phi(s_i', s_{-i})$$
$$= u_i(s_i, s_{-i}) - u_i(s_i', s_{-i}).$$

Assuming $|S_i| < \infty \quad \forall i \in N$

Theorem 23.08.2024-1: Every potential game has a PSNE.

Proof: Suppose that s^* is a strategy profile that maximizes ϕ.

$$\therefore \phi(s_i^*, s_{-i}^*) \geq \phi(s_i', s_{-i}^*) \quad \forall i \in N, \forall s_i' \in S_i$$

$$\Rightarrow u_i(s_i^*, s_{-i}^*) \geq u_i(s_i', s_{-i}^*) \quad \forall i \in N, \forall s_i' \in S_i$$

$\Rightarrow s_i^*$ is a PSNE. $\square$

Best Response Dynamics (BRD):

- Pick an arbitrary strategy profile s
- While s is not a PSNE
  - * find i, $s_i' \in S_i$ s.t. $u_i(s_i', s_{-i}) > u_i(s)$
  - * replace s_i with s_i' in s.

This terminates because we keep increasing the potential function at each step. Since potential function can take only finitely many values, this infact terminates.

BRD converges to PSNE in every (finite) potential game

Consider the graph $G = (V, E)$,

V: Set of strategy profiles.

$(s, s') \in E$ iff $\exists i, \exists s_i' \in S_i$ s.t. $u_i(s_i', s_{-i}) > u_i(s)$

where $s = (s_i, s_{-i})$, $s' = (s_i', s_{-i})$

- Such a graph would be a DAG for potential games, and BRD terminates at some node with some outdegree = 0 node. Potential increases at each step.
- For games other than potential games, this graph might not be acyclic, and BRD can fall into a cyclic loop.
- Can take exponential-in-N time.

Example: Network-Congestion-Game Revisited

$G = (V, E)$, Player i: (s_i, t_i).

S_i: Set of $s_i - t_i$ paths.

$P = (P_1, P_2, \ldots, P_n)$ strategy profiles.

load $l_e(P) = |\{i \mid e \in P_i\}|$

congestion function $c_e : \{0, 1, \ldots, n\} \to \mathbb{R}$

$$u_i(P) = \sum_{e \in P_i} c_e(l_e(P)).$$

This is a potential game.

$$\phi(P) = \sum_{e \in G} \sum_{j=1}^{l_e(P)} c_e(j)$$

To show: let $P = (P_1, P_2, \ldots, P_i, \ldots, P_n)$

$\hat{P} = (P_1, P_2, \ldots, \hat{P}_i, \ldots, P_n)$

then $\phi(\hat{P}) - \phi(P) = \sum_{e \in \hat{P}_i} c_e(l_e(\hat{P})) - \sum_{e \in P_i} c_e(l_e(P))$.

Proof:

$$\phi(\hat{P}) = \sum_{e \in G} \sum_{j=1}^{l_e(\hat{P})} c_e(j)$$

$$= \sum_{e \in G \setminus (P_i \cup \hat{P}_i)} \sum_{j=1}^{l_e(P)} c_e(j) + \sum_{e \in P_i \cap \hat{P}_i} \sum_{j=1}^{l_e(P)} c_e(j)$$

$$+ \sum_{e \in P_i \setminus \hat{P}_i} \sum_{j=1}^{l_e(\hat{P})} c_e(j) + \sum_{e \in \hat{P}_i \setminus P_i} \sum_{j=1}^{l_e(\hat{P})} c_e(j)$$

Similarly:

$$\phi(P) = \sum_{e \in G} \sum_{j=1}^{le(P)} c(j)$$

$$= \sum_{e \in G \setminus (P_i \cup \hat{P}_i)} \sum_{j=1}^{le(P)} c(j) + \sum_{e \in P_i \cap \hat{P}_i} \sum_{j=1}^{le(P)} c(j)$$

$$+ \sum_{e \in P_i \setminus \hat{P}_i} \sum_{j=1}^{le(P)} c(j) + \sum_{e \in \hat{P}_i \setminus P_i} \sum_{j=1}^{le(P)} c(j)$$

Loads in edges not in P_i or $\hat{P}_i$ do not change. Moreover, loads of edges in both P_i as well as $\hat{P}_i$ do not change.

$$\therefore \sum_{e \in G \setminus (P_i \cup \hat{P}_i)} \sum_{j=1}^{le(P)} c(j) = \sum_{e \in G \setminus (P_i \cup \hat{P}_i)} \sum_{j=1}^{le(\hat{P})} c(j)$$

and $$\sum_{e \in P_i \cap \hat{P}_i} \sum_{j=1}^{le(P)} c(j) = \sum_{e \in P_i \cap \hat{P}_i} \sum_{j=1}^{le(\hat{P})} c(j)$$

As $le(P) = le(\hat{P}) \; \forall e \in (G \setminus (P_i \cup \hat{P}_i)) \cup (P_i \cap \hat{P}_i)$.

For $e \in \hat{P}_i \setminus P_i$, $le(\hat{P}) = le(P) + 1$

and $e \in P_i \setminus \hat{P}_i$, $le(\hat{P}) = le(P) - 1$.

$$\therefore \Phi(\hat{P}) - \Phi(P)$$

$$= \sum_{e \in \hat{P}_i \setminus P_i} c_e(l_e(\hat{P})) - \sum_{e \in P_i \setminus \hat{P}_i} c_e(l_e(P)).$$

$$= \left(\sum_{e \in \hat{P}_i \setminus P_i} c_e(l_e(\hat{P})) + \sum_{e \in P_i \cap \hat{P}_i} c_e(l_e(\hat{P})) \right)$$

$$- \left(\sum_{e \in P_i \setminus \hat{P}_i} c_e(l_e(P)) + \sum_{e \in P_i \cap \hat{P}_i} c_e(l_e(P)) \right)$$

$$= \sum_{e \in \hat{P}_i} c_e(l_e(P)) - \sum_{e \in P_i} c_e(l_e(P)).$$

□.

Coming back to BRD... We can either wait for a long time while it runs or settle for an "approximate PSNE".

Approximate PSNE:

ε-PSNE:

For $\Gamma = (N, (S_i)_{i \in N}, (u_i)_{i \in N})$, $s = (s_1, \ldots, s_n)$ is an ε-PSNE if $\forall i \in N$, $\forall s'_i \in S_i$

$$u_i(s'_i, s_{-i}) \leq (1+\varepsilon) u_i(s).$$

To capture this we have ε-BRD.

ε-BRD:

- Pick an arbitrary strategy profile s
- While s is not an ε-PSNE
  - find i, s_i' s.t. $u_i(s_i', s_{-i}) > (1+\varepsilon)\, u_i(s)$
  - replace s_i with s_i' in s.

Network Congestion Games Revisited

n players; $G = (V, E)$

player i: (src_i, snk_i) $\quad S_i$: set of all paths from src_i to snk_i

$P = (P_1, P_2, \ldots, P_n)$, $\quad \ell_e(P) = |\{i : e \in P_i\}|$

$c_e : \{0, 1, \ldots, n\} \longrightarrow \mathbb{R}$ (non-decreasing)

$$- u_i(P) = C_i(P) = \sum_{e \in P_i} c_e(\ell_e(P))$$

$$C(P) = \sum_{i=1}^{n} C_i(P) = \sum_{e \in P_i} \ell_e(P) \cdot c_e(\ell_e(P)).$$

Goal: minimize cost.

A strategy profile s is an ε-PSNE if

$$\forall i \in [n], \forall s_i' \in S_i \quad C_i(s_i', s_{-i}) \geq (1-\varepsilon)\, C_i(s).$$

Cost Minimization:

$S = S_1 \times \cdots \times S_n$, S_i: strategy of player i. $s \in S$

$C_i(s)$: cost incurred by player i on s. $C(s)$: Total cost.

Goal of player i: minimize $C_i(s)$.

In any finite potential game. ε-BRD converges to an ε-PSNE.

Max-Gain Best Response:

Among all players with an ε-move, choose the one that lead to largest decrease in absolute cost.

Theorem 29.08.2024-1: Consider a Network Congestion Game, with:

- [Symmetry] All players have the same source vertex, all players have the same sink.
- Cost function satisfies 'α-bounded jump' condition; ($\alpha \geq 1$)

$$c_e(x+1) \in [c_e(x), \alpha \cdot c_e(x)] \quad \forall x, e$$

- Max-Gained variant of ε-BRD is used

then ε-PSNE is reached in

$$O\left(\frac{n\alpha}{\varepsilon} \log \frac{\Phi(P_0)}{\Phi_{min}}\right) \text{ iterations}$$

where P_0: strategy profile ~~of the ini~~ with which ε-BRD starts.

Φ_{min}: Minimum value of the potential function.

$$\Phi(P) = \sum_{e \in G} \sum_{j=1}^{l_e(P)} c_e(j).$$

Proof:

Lemma 29.08.2024-2: In every strategy profile P, $\exists$ player i with $C_i(P) \geq \frac{\phi(P)}{n}$

Proof:

$$\phi(P) = \sum_{e \in G} \sum_{j=1}^{l_e(P)} c_e(j)$$

$$\leq \sum_{e \in G} \sum_{j=1}^{l_e(P)} c_e(l_e(P)) \quad , \left[c_e \text{ is non-decreasing}\right]$$

$$= \sum_{e \in G} l_e(P)\, c_e(l_e(P)) = C(P) = \sum_{i=1}^{n} C_i(P)$$

$$\therefore \max_{i=1}^{n} \{C_i(P)\} \geq \frac{C(P)}{n} \geq \frac{\phi(P)}{n}$$.

Lemma 29.08.2024-3: Suppose player i is chosen in Max-Gain ε-BRD and i makes a move from P_i to P_i'. Then $C_i(P) - C_i(P_i', P_{-i}) \geq \frac{\varepsilon}{\alpha} C_j(P)$, for every other player j. for α-bounded jump c_e, & game is symmetric.

Proof: Fix j.

Case I: j has an ε-move

Because i is chosen, we must have

$$C_i(P) - C_i(P_i', P_{-i}) \geq C_j(P) - C_j(P_j'', P_{-j}), \forall P_j'' \in S_j$$

$$\geq \varepsilon\, C_j(P) \geq \frac{\varepsilon}{\alpha} C_j(P).$$

Case 2: j does not have an ε-move.

Since the game is symmetric, $S_i = S_j$.

$\therefore P_i \to P_i'$ is ε move for i but
$P_j \to P_i'$ is not an ε move for j.

$\therefore$ ① $C_i(P_i', P_{-i}) < (1-\varepsilon)\, C_i(P)$

② $C_j(P_i', P_{-j}) \geq (1-\varepsilon)\, C_j(P)$

$(P_i', P_{-i}) = (P_1, \ldots, P_i', \ldots, P_j, \ldots, P_n)$

$P = (P_1, \ldots, P_i, \ldots, P_j, \ldots, P_n)$

$(P_i', P_{-j}) = (P_1, \ldots, P_i, \ldots, P_i', \ldots, P_n)$

So the collection of paths (P_i', P_{-i}) and (P_i', P_{-j}) differ by one path; hence they have $(n-1)$ common strategies. Hence the load on every edge differs by at most 1. Therefore by α bounded jump condition.

③ $C_j(P_i', P_{-j}) = \sum_{e \in P_i'} c_e(l_e(P_i', P_{-j}))$

$\leq \alpha \sum_{e \in P_i'} c_e(l_e(P_i', P_{-i}))$

$= \alpha\, C_i(P_i', P_{-j})$

$\therefore$ ①, ②, ③ are compatible only if :

$$(1-\varepsilon)C_j(P) \overset{②}{\leq} C_j(P_i', P_{-j}) \overset{③}{\leq} \alpha C_i(P_i', P_{-i}) \overset{④}{<} \alpha(1-\varepsilon)C_i(P)$$

$$\Rightarrow C_j(P) < \alpha C_i(P)$$

$\therefore$ Combining with ①, this gives :

$$C_i(P) - C_i(P_i', P_{-i}) > \varepsilon C_i(P) \geq \frac{\varepsilon}{\alpha} C_j(P)$$

Now we prove our original theorem.

Proof of Theorem 29.08.2024-1:

Let player i make a move from P_i to P_i'.

$$\therefore \phi(P) - \phi(P_i', P_{-i}) = C_i(P) - C_i(P_i', P_{-i})$$

$$\geq \max_j \left\{\frac{\varepsilon}{\alpha} C_j(P)\right\}, \left[\text{Lemma } 29.08.2024\text{-}2\right]$$

$$\geq \frac{\varepsilon}{\alpha} \frac{\phi(P)}{n}, \left[\text{Lemma } 29.08.2024\text{-}1\right]$$

$\therefore$ ϕ drops by a (multiplicative ~~factor~~) of factor of $\frac{\varepsilon}{\alpha n}$.

$$\therefore \phi(P_i', P_{-i}) \leq \left(1 - \frac{\varepsilon}{n\alpha}\right)\phi(P).$$

$\therefore$ After $\frac{\alpha n}{\varepsilon}$ many iterations, ϕ drops by a factor

$$\leq \left(1 - \frac{\varepsilon}{n\alpha}\right)^{\frac{\alpha n}{\varepsilon}} \leq \left(e^{-\frac{n\alpha}{\varepsilon}}\right)^{\frac{\alpha n}{\varepsilon}} = e^{-1} = \frac{1}{e}.$$

So in $\log_e\left(\frac{\phi(P_0)}{\phi_{min}}\right)$ - many chunks of $\frac{n\alpha}{\varepsilon}$ iterations, potential has to drop to minimum.

∴ The algorithm terminates & reports an ε-PSNE in

$$O\left(\frac{n\alpha}{3\varepsilon}\log_e\frac{\phi(P_0)}{\phi_{min}}\right)$$

iterations. ▨

Polynomial Local Search (PLS): a new complexity class

The MAX-CUT problem:

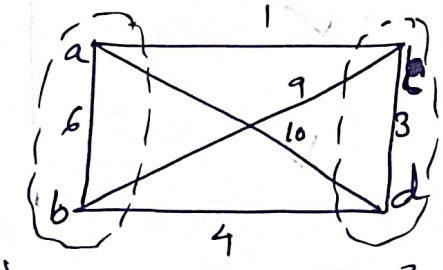
Input: $G=(V,E)$ $\forall e \in E$, $w_e \geq 0$

Task: Partition $V = S \cup \bar{S}$. s.t. $\sum_{\substack{u \in S \\ v \in \bar{S}}} w_{uv}$ is maximized.

Local search algorithm for MAX-CUT:

- Start with arbitrary cut $S, \bar{S}$.
- If $\exists u \in S$ s.t. the cut $S \setminus \{u\}, \bar{S} \cup \{u\}$ increases the total weight, shift u to $\bar{S}$.
- If $\exists v \in \bar{S}$ s.t. the cut $S \cup \{v\}, \bar{S} \setminus \{v\}$ increases the total weight, shift v to S.

Example:



Local Optimum (cost = 24)

Global optimum is $\{a, c\}, \{b, d\}$ with cost 28

* If $w_e = 1$ $\forall e \in E$, Local search terminates in polytime
* In general, Local search terminates in $O(\sum_{e \in E} w_e)$ time; which may not be ~~optimum~~ polynomial in $|V|, |E|$.

Detour: Congestion Games:

E: set of resources.

n players

$c_e(i)$ $\forall e \in E$ $i \in \{1, 2, \ldots, n\}$

strategy set for player i, $S_i \subseteq 2^E$

Potential function $\phi(s) = \sum_{e \in E} \sum_{i=1}^{n_e(s)} c_e(i)$

$n_e(s)$: # strategies in s using resource e.

	Network Congestion games	Congestion.
• Convergence of BRD (ε-BRD) to PSNE (ε-PSNE)	✓	✓
• Polytime convergence of ε-PSNE (with restriction)	✓	✓
• Existence of symmetric version where PSNE is found polytime.	✓	✗ (believed)

PLS Revisited

An "abstract local search problem" is specified by three polytime algorithms:

(1) Takes as input an instance and outputs a feasible solution.

(2) Takes as input an instance & a feasible solution & outputs objective function value of the input solution.

(3) Takes as input & an instance & a feasible solution, outputs 'LOCALLY-OPTIMAL' or an improved solution.

Local Search Algorithm:

- Run algo (1) to find solution s
- Repeat until ~~s is locally optimum~~ s is locally optimal.
  $s \leftarrow$ algo (3) on i/p.

There are finitely many feasible solutions (because algo (2) can take at most exponentially many feasible instance to be able to read them & output something in polytime). Therefore any local search algorithm terminates in finite time ~~(pay~~ (may not be polytime).

PLS (Polynomial Local Search): Set of all local search problems. Clearly $PLS \supseteq P$. Belief $P \neq PLS$

PLS-Completeness:

Reduction from $L_1 \in PLS$ to $L_2 \in PLS$ is a pair of polytime algorithms (A, B) s.t.

* A maps instance $x \in L_1$ to $A(x) \in L_2$
* B maps a local optimum for $A(x)$ to a local optimum for x.

~~$L \in PLS$ is PLS-complete~~

$L \in PLS$ is PLS-Complete if L' reduces to L $\forall\, L' \in PLS$.

Here P is the set of problems whose local optimum can be found in polytime. So Local optimum for unweighted maxcut is in P.

The Best we can hope for is running the ~~brute force~~ local search algo.

Fact: Finding local optimum for a MAX-CUT instance with general non-negative weights is PLS-Complete

CG
Problem of computing PSNE for congestion games.

Theorem 30.08.2024-1: CG is PLS-Complete.

Proof: Let the general max-cut's local optimal problem be named MC.

Part 1: CG ∈ PLS:

(1) Pick any strategy from S_i, $\forall S_i$
(2) $\phi(s)$ can be computed as, in definition, in polytime.
(3) on input s, run one iteration of BRD & output the resulting strategy profile.

PSNE's are local optimums of CG.

Part 2: We reduce $MC \leq CG$:

Let $G(V, E)$, $\{w_e\}_{e \in E}$ be an ~~any~~ MC instance ($w_e \geq 0 \ \forall e \in E$).

Construct an instance of CG as follows:

- set of players : V
- for every edge $e \in E$ include two resources $r_e, \bar{r}_e$.
- for $v \in V$, S_v contains exactly two strategies:

$$\{r_e\}_{e \in \delta(v)} \text{ and } \{\bar{r}_e\}_{e \in \delta(v)}$$

$\delta(v) :=$ set of edges incident on v.

- cost of r_e or $\bar{r}_e$ is 0 if one player uses it, and w_e if at least two player uses it.

Notice that any resources can be used by at most two (endpoints of an edge).

$\therefore r_e/\bar{r}_e$ can be used by ~~at most~~ 0, 1, 2 players.

The combined load on r_e and $\bar{r}_e$ in any strategy profile is exactly 2.

A vertex $u \in S \leftrightarrow u$ uses $\{r_e\}_{e \in \delta(u)}$

A vertex $u \in \bar{S} \leftrightarrow u$ uses $\{\bar{r}_e\}_{e \in \delta(u)}$.

Cuts of G with weight $w(S, \bar{S})$ correspond to strategy profile of congestion game with potential

$$\phi(s) = \sum_{e \in E} w_e - w(S, \bar{S})$$

edges in S contribute w_e
edges in S' contribute w_e
edges crossing S, S' contribute 0

Higher the weight of cut, lower the value of $\phi(s)$.

∴ Local maximum of MC ↔ PSNE in CG

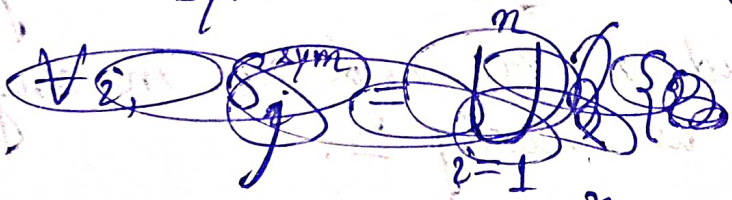
∴ CG is PLS-Complete □

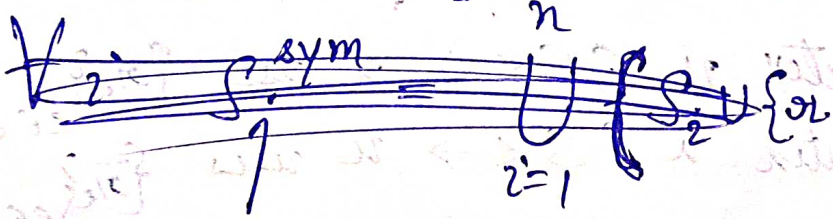
Theorem 30.08.2024-2: The symmetric version of CG, CG_{sym} is also PLS complete.

Proof: We show $CG \leq CG_{sym}$.

Include n new resources $r_1, \ldots, r_n$.

$E_{sym} = E \cup \{r_1, \ldots, r_n\}$.





$$\forall j,\ S_j^{sym} = S_{sym} = \bigcup_{i=1}^{n} \{ s \cup \{r_i\} \mid \forall s \in S_i \}.$$

We want to define the cost function, such that PSNE in CG_{sym} $\leftrightarrow$ PSNE in CG.

cost of all resources in E remain the same.
cost of resource r_i is 0 if 1 player uses it, and ∞ if 2 or more player uses it.

Therefore, clearly PSNE in CG $\rightarrow$ PSNE in CG_{sym} (take same strategy profile).

~~Now, PSNE of~~

Now, in a PSNE of CG_{sym}, every player takes a different r_i (to make cost $< \infty$).

$\therefore$ i-th player in CG chooses the strategy s, where $s \cup \{r_i\}$ is chosen by some player in CG_{sym}.

Since CG_{sym} is PSNE, no other $s' \cup \{r_i\}$ would give a better cost $\Rightarrow$ no other s' for i-th player in CG would give a better cost $\Rightarrow$ CG also has this PSNE.

$\therefore$ CG_{sym} is PLS-Complete.

MSNE for Bimatrix Games

Player 1 (A) and Player 2 (B).

(x^*, y^*) is an MSNE for the game if

$$x^* A y^{*T} \geq x A y^{*T} \quad \forall x$$

and $$x^* B y^{*T} \geq x^* B y^T \quad \forall y.$$

No polytime algorithms are known to find MSNE in bimatrix games.

We wish to prove some formal hardness results on this.

Detour: Versions of a problem:

- **Decision**: Given $n \in \mathbb{Z}^+$, is there an integer k, $1 < k < n$ s.t. $k | n$

  (This is in NP)

- **Search**: Given $n \in \mathbb{Z}^+$, find k, $1 < k < n$ s.t. $k | n$.

  (This is in Functional NP or FNP)

Functional NP (FNP): Consists of 'search' versions of problems in NP.

FNP-Completeness:

$L_1 \in$ FNP reduces to $L_2 \in$ FNP if there exists a poly-time algorithm:

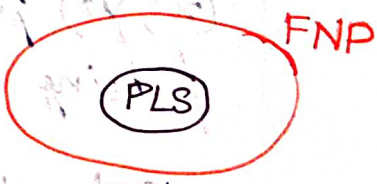
- A maps $x \in L_1$ to instance $A(x) \in L_2$
- B maps a solution/witness for $A(x)$ to a solution/witness for x.

FNP-Complete problems are problems L s.t $\forall L' \in FNP$, L' reduces to L ; and $L \in FNP$.

Examples of FNP-Complete problems

- Given a graph G, find a path that visits every vertex exactly once. (HAM-PATH)

Note that $PLS \subseteq FNP$ (because any solution to PLS can be verified using the poly-time algorithms used to define PLS).



Let

$MSNE_B$: Problem of computing MSNE in bimatrix games.

Is $MSNE_B$ an FNP complete problem ?

Firstly $MSNE_B \in FNP$

Theorem 31.08.2024-1: If $MSNE_B$ is FNP-Complete, then $NP = coNP$.

Proof: Let us assume that $MSNE_B$ is FNP-Complete.

$\Rightarrow$ $\underbrace{SAT}_{\text{search version}} \leq_p MSNE_B$

[P.T.O]

$\Rightarrow \exists$ polytime algorithms A and B, s.t.

- A maps formula ϕ to an instance (C, D) of $MSNE_B$
- B maps a mixed strategy for game (C, D) to a satisfying assignment for ϕ. (or "No" if B decides ϕ is not satisfiable).

Now, what happens if ϕ is unsatisfiable? Then applying ~~A, B sequentially~~, A gives an instance of $MSNE_B$. Nash's theorem guarantees that there exists MSNE (x^*, y^*) for (C, D). Now applying B on (x^*, y^*) provides a short witness for ϕ being unsatisfiable

$\Rightarrow$ UNSAT $\in$ NP

But UNSAT $\in$ coNP-Complete

$\Rightarrow$ NP = coNP. □

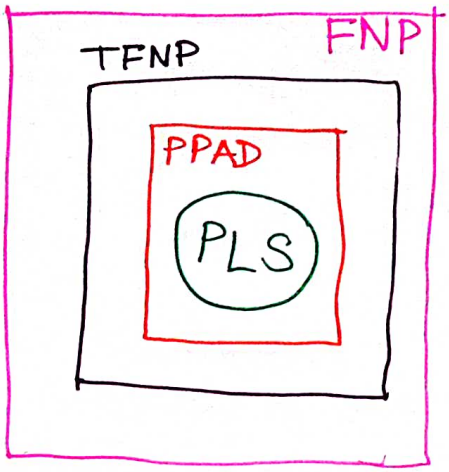
(Total FNP)

TFNP: Problems in FNP where every instance has at least one witness.

By Nash's Theorem $MSNE_B \in$ TFNP.

We will also define something called PPAD and show part of the proof for

$MSNE_B \in$ PPAD-Complete.



The definition of TFNP is rather vague and semantic; so we cannot really define completeness

PPAD (Polynomial Parity Arguments directed versions):

Consists of problems solvable by a particular class of naive-directed path following algorithms.

PLS ⊆ PPAD (local search can also be thought of as a path-following algorithm).

[PLS: Computing a sink vertex in a DAG]

PPAD:

In directed graphs with ≥1 source nodes, ∃ polytime algorithms

1. Pick an initial solution
2. Output that a current solution is a witness or compute & output the next intermediate solution.

Notice that there are no objective functions.

We will see that $MSNE_B$ is PPAD-complete.

Correlated Equilibrium:

$$\Gamma = \left(N, (S_i)_{i\in N}, (u_i)_{i\in N}\right); \quad S = S_1 \times S_2 \times \cdots \times S_n$$

σ, a distribution on S is a Correlated Equilibrium (CE), if $\forall i \in N$, $\forall s_i \in S_i$ and $\forall s_i' \in S_i$

$$\mathop{\mathbb{E}}_{s\sim\sigma}\left[u_i(s) \mid s_i\right] \geq \mathop{\mathbb{E}}_{s\sim\sigma}\left[u_i(s_i', s_{-i}) \mid s_i\right]$$

equivalently,

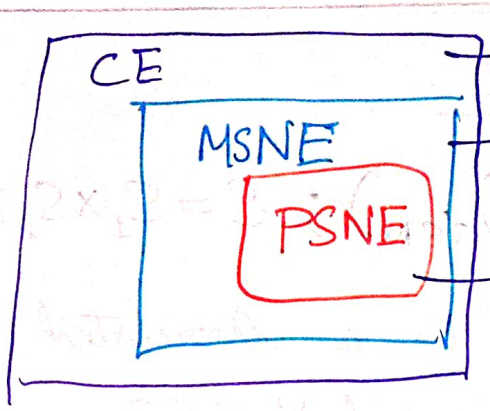
$$\sum_{s\in S} \sigma(s \mid s_i) \cdot u_i(s) \geq \sum_{s\in S} \sigma(s \mid s_i)\, u_i(s_i', s_{-i}).$$

Interpretation:

- Suppose there is a trusted third party T
- σ is known publically
- T chooses $s \xleftarrow{\sigma} S$
- $\forall i \in N$, T communicates (privately) s_i to player i.
- Player i may choose to deviate (play s_i or play something else)
- At this point, player i knows σ, s_i and hence a ~~point~~ posterior distribution on s_{-i}.

σ is a CE, if no player can improve their expected utility by deviating from s_i, conditioned on the information known.

$(\sigma_1, \sigma_2, \ldots, \sigma_n)$ is an MSNE, then the product distribution is a CE.



Solvable using LP (size of LP may be exponential).

Always exist, difficult to compute

May not exist

Example (consider cars at a an intersection)

		[Player 2] Stop	[Player 2] Go
[Player 1]	Stop	(0, 0)	(0, 1)
	Go	(1, 0)	(-10, -10)

PSNE: ~~(1, 0), (0~~ (Go, Stop), (Stop, Go)

MSNE: ?

CE: (Stop, Stop), (Stop, Go), (Go, Stop), (Go, Go)

0 , 1/2 , 1/2 , 0

Expected Payoff: 1/2

(Note that (0, 1/2, 1/2, 0) cannot be expressed as a product distribution.)

~~There exists a class of problems where there are learning algorithms which~~

Coarse Correlated Equilibrium (CCE)

There exists a class of learning algorithms for a faster computation.

* Also computable with LP

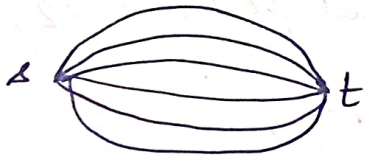
σ, a distribution on S is a coarse correlated equilibrium (CCE) if $\forall i \in N$, $\forall s_i, s_i' \in S_i$

$$\mathop{\mathbb{E}}_{s \sim \sigma}[u_i(s)] \geq \mathop{\mathbb{E}}_{s \sim \sigma}[u_i(s_i', s_{-i})]$$

The inequation must also hold true when s_i is known. Hence every CCE is a CE (Formal proof: Exercise)

Example:

Consider the following instance to a Network Congestion Game, $N = \{1, 2, 3, 4\}$.



$E = \{0, 1, \ldots, 5\}$. $c(x) = x$.

u_i = # players choosing edge j (when i plays $j \in E$).

PSNE: All profiles with edges are distinct.

($\binom{6}{4}$ profiles out of 6^4)

MSNE: All players choose ~~are~~ a uniform random edge. (formally prove).

Expected cost = $3/2$ (prove).

CE:

A = set of all strategy profiles where 2 players choose 2 distinct edges while the remaining 2 players choose a third edge.

uniform distribution on $A \longrightarrow$ CE (not an MSNE).

expected payoff $= 3/2$

Exercise: verify

CCE:

Consider $A' \subseteq A$ to be the set of profiles where the set of edges is either $\{1, 3, 5\}$ or $\{0, 2, 4\}$

uniform distribution on A' is a CCE

(expected payoff $= 2/2$)

(This is NOT a CE). Exercise: verify

External Regret Framework

There is a single player P and an adversary Adv.
At time $t = 1, 2, \ldots, T$

- P picks ~~up front~~ probability distribution p_t over set of actions A
- Adv picks up a utility function $\pi_t : A \to [0, 1]$.
- P chooses action $a_t \xleftarrow{p_t} A$ and obtains payoff $\pi_t(a_t)$

Note that Adv has access to $p_t, p_{t-1}, \ldots, p_1, \pi_{t-1}, \pi_{t-2}, \ldots, \pi_1$ (adaptive adversary)

B: Benchmark (depends on π_t)

Time averaged regret of the sequence $a_1, \ldots, a_t$ w.r.t action a, is:

$$\frac{1}{T}\left(B - \sum_{t=1}^{T} \sum_{a_t \in A} p_t(a_t) \cdot \pi_t(a_t)\right)$$

} We want this $\to 0$ when $T \to \infty$.

Choice of B:

Suppose we choose benchmark to "the best action possible".

$$\sum_{t=1}^{T} \max_{a_t \in A} \pi_t(a_t).$$

This is not a good benchmark. This is because of the following example.

$|A| = 2 \qquad p_t = (1/2, 1/2) \longrightarrow \pi_t = (1, 0)$

Expected Payoff $\leq T$
But $\sum_{t=1}^{T} \max_{a_t \in A} \pi_t(a_t) = T$

} Time avg regret $\geq \frac{1}{T}\left(T - \frac{T}{2}\right) = 1/2$

* How about Best fixed action.

$$\max_{a_t \in A} \sum_{t=1}^{T} \pi_t(a_t)$$

w.r.t. best fixed action benchmark, there is a no-regret algorithm.

No regret: ~~regret~~ regret $\to 0$ when $T \to \infty$.

No no-regret algorithm can be deterministic.

Multiplicative ~~update~~ Weights (MW) algorithm:

$w_0(a) = 1 \quad \forall a \in A$

for $t = 1, \ldots, T$; do:

- Play an ~~action~~ action a with probability $w_{t-1}(a) / \Gamma_{t-1}$
- Given $\pi_t(\cdot)$, ~~define~~ define

$$w_t(a) = w_{t-1}(a)(1+\epsilon)^{\pi(a)} \quad \forall a \in A$$

~~(where $\Gamma_t = \sum_{a \in A} w_t(a)$. $\forall a \in A$.~~

where $\Gamma_t = \sum_{a \in A} w_t(a)$.

Theorem 27-09-2001-1: Time averaged regret of MW algorithm is $O\left(\sqrt{\frac{\ln n}{T}}\right)$ for appropriate choice of ϵ.

where $n = |A|$

Proof:

$$OPT = \max_{a \in A} \sum_{t=1}^{T} \pi_t(a) \quad \because \quad \sum_{t=1}^{T} \sum_{a \in A} p_t(a) \pi_t(a)$$

Let a^* be the action $a^* \in A$ for which,

$\sum_{t=1}^{T} \pi_t(a_t)$ is ~~maximum~~ maximum.

$\therefore \quad \Gamma_T \geq w_T(a^*)$

$= w_0(a^*) \prod (1+\varepsilon)^{\pi_t(a^*)}$

$= (1+\varepsilon)^{\sum_{t=1}^{T} \pi_t(a^*)} = (1+\varepsilon)^{OPT}$ ——— (i)

Payoff in t^{th} step

$$\sum_{a \in A} \pi_t(a_t) p_t(a_t)$$

$$= \sum_{a \in A} \pi_t(a) \frac{w_{t-1}(a)}{\Gamma_{t-1}} = v_t$$

$$\therefore \Gamma_T = \sum_{a \in A} w_T(a)$$

$$= \sum_{a \in A} w_{T-1}(a) (1+\varepsilon)^{\pi_T(a)}$$

$$\leq \sum_{a \in A} w_{T-1}(a) (1+\varepsilon \pi_T(a)) \quad [\text{as } \pi_t \in [0,1]]$$

$$= \sum_{a \in A} w_{T-1}(a) + \varepsilon \sum_{a \in A} w_{t-1}(a) \pi_T(a)$$

$$= \Gamma_{T-1} + \varepsilon \Gamma_{T-1} v_T = \Gamma_{T-1}(1+\varepsilon v_T)$$

$$\leq \text{[illegible]}\, \Gamma_0 \prod_{t=1}^{T} (1+\varepsilon v_t)$$

$$= n \prod_{t=1}^{T} (1+\varepsilon v_t) \qquad \text{——— (ii)}$$

∴ From (i) and (ii), we get.

$$(1+\varepsilon)^{OPT} \leq \Gamma_T \leq n \prod_{t=1}^{T} (1+\varepsilon v_t)$$

$$\Rightarrow OPT \ln(1+\varepsilon) \leq \ln \Gamma_T \leq \ln n + \sum_{t=1}^{T} \ln(1+\varepsilon v_t)$$

Now, using $x - x^2 \leq \ln(1+x) \leq x \;\forall x \in [0,1]$:

$$OPT(\varepsilon - \varepsilon^2) \leq \ln n + \varepsilon \sum_{t=1}^{T} v_t$$

$$\Rightarrow OPT(1-\varepsilon) \leq \frac{\ln n}{\varepsilon} + \sum_{t=1}^{T} v_t$$

$$\Rightarrow OPT - \sum_{t=1}^{T} v_t \leq \varepsilon OPT + \frac{\ln n}{\varepsilon}$$

$$\leq \varepsilon \cdot T + \frac{\ln n}{\varepsilon}, \quad [OPT \leq T]$$

$$\therefore \frac{1}{T}\left(OPT - \sum_{t=1}^{T} v_t\right) \leq \varepsilon + \frac{\ln n}{\varepsilon T}.$$

if $\varepsilon = \sqrt{\frac{\ln n}{T}}$, we have regret $\leq 2\sqrt{\frac{\ln n}{\varepsilon T}}$ □

Note that as $T \longrightarrow \infty$, regret $\to 0$. (No-regret)

Correlated Equilibrium Revisited:

Alternate definition: σ is a CE if for every switching function $\delta : S_i \to S_i$

$$\mathop{\mathbb{E}}_{s \sim \sigma}[u_i(s)] \geq \mathop{\mathbb{E}}_{s \sim \sigma}[u_i(\delta(s_i), s_{-i})].$$

Exercise: show equivalence with the original definition.

External Regret Framework Revisited (single player, n actions)

We saw that:

There is a no regret algorithm (MW algorithm) whose expected time-averaged regret is

$$O\left(\sqrt{\frac{\log n}{T}}\right)$$

Note that the algorithm we saw knew (and used) the value of T. We will see what happens if we do not know that.

Unknown Time Horizon (T):

Claim 03.10.2024-1: $\exists$ a no-regret algorithm whose expected time-averaged regret is $O\left(\sqrt{\frac{1}{T}\log n}\right)$ w.r.t. every fixed action.

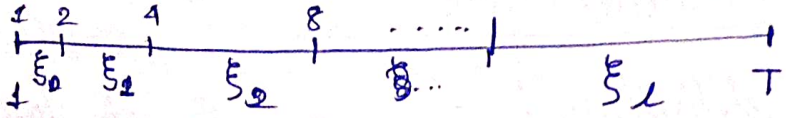
Proof: We modify the MW ~~is~~ as follows:

set ε ~~to~~ at time t to $\sqrt{\frac{\ln n}{T_t}}$.

T_t: smallest power of 2 $\geq t$

If t is a power of 2, reset $w_t(a) = 1 \ \forall a \in A$.

We now analyse this algorithm.



We partition $[1, T]$ into l epochs:

$$\xi_i = [2^i, \min\{2^{i+1}, T\}]$$

Let $OPT_i := \max_{a \in A} \sum_{t \in \xi_i} \pi_t(a)$.

and $OPT = \max_{a \in A} \sum_{t=1}^{T} \pi_t(a)$.

We showed that

$$OPT - \sum_{t=1}^{T} v_t \leq 2\sqrt{T \log n}$$

where $v_t = \sum_{a \in A} \pi_t(a) \cdot \frac{w_{t-1}(a)}{\Gamma_{t-1}}$; $\Gamma_{t-1} = \sum_{a \in A} w_{t-1}(a)$.

Now, clearly $OPT \leq \sum_{i=0}^{l} OPT_i$, because sum-of-max $\geq$ max-of-sum.

$$\therefore \left(OPT - \sum_{t=1}^{T} v_t\right) \leq \sum_{i=0}^{l} \left(OPT_i - \sum_{t \in \xi_i} v_t\right)$$

Now, we are running the MW algorithm afresh for each epoch.

$$\therefore \left(OPT - \sum_{t=1}^{T} v_t\right) \leq \sum_{i=0}^{l} \left(OPT_i - \sum_{t \in \xi_i} v_t\right)$$

$$\leq \sum_{i=0}^{l} 2\sqrt{2^i \ln n}$$

$$\leq 2\sqrt{\ln n} \sum_{i=0}^{\ell} 2^{i/2}$$

$$\leq 2\sqrt{\ln n} \cdot 2^{\ell/2+1}$$

$$= 4\sqrt{2^{\ell} \ln n}$$

$$\leq 4\sqrt{T \ln n}.$$

$$\therefore \frac{1}{T}\left(OPT - \sum_{t=1}^{T} v_t\right) \leq 4\sqrt{\frac{\ln n}{T}}.$$

□

No Regret Dynamics:

This is a kind of a learning algorithm.

- n players
- set of actions for i : S_i
- player i chooses $p_i \in \Delta(S_i)$ using the no-regret (any) algo.
- payoff π_i given to player i
  $=$ expected utility $u_i(s_i, p_{-i})$:
  if i plays s_i & every other player $j \in N \setminus \{i\}$ plays according to dist. p_j.

$$= \sum_{s_{-i} \in S_{-i}} u_i(s_i, s_{-i}) \prod_{j \neq i} p_j(s_j).$$

This was one iteration of the learning algorithm.

Theorem 03.10.2024-2: For $\varepsilon > 0$, suppose after T iterations, time-averaged regret of every player $< \varepsilon$. Let $\sigma_t = \prod_{i=1}^{n} p_{t,i}$ be the product distribution of t-th iteration for $t \in [T]$, and $\sigma = \frac{1}{T} \sum_{t=1}^{T} \sigma_t$ be the time averaged distribution. Then σ is an ε-CCE for the game.

Definition (ε-CCE): σ is an ε-CCE if

$$\mathop{\mathbb{E}}_{s \sim \sigma} [u_i(s)] \geq \mathop{\mathbb{E}}_{s \sim \sigma} [u_i(s_i', s_{-i})] - \varepsilon \quad \forall s_i' \in S_i$$

Proof of Theorem 03.10.2024-2:

$$\mathop{\mathbb{E}}_{s \sim \sigma} [u_i(s)] = \frac{1}{T} \sum_{t=1}^{T} \mathop{\mathbb{E}}_{s \sim \sigma_t} [u_i(s)], \quad \left[\text{as } \sigma = \frac{1}{T} \sum_{t=1}^{T} \sigma_t\right]$$

now, $\forall s_i' \in S_i$, look at:

$$\mathop{\mathbb{E}}_{s \sim \sigma_t} [u_i(s_i', s_{-i})] - \mathop{\mathbb{E}}_{s \sim \sigma_t} [u_i(s)] \leq \max_{s_i' \in S_i} \left(\mathop{\mathbb{E}}_{s \sim \sigma_t} [u_i(s_i', s_{-i})] - \mathop{\mathbb{E}}_{s \sim \sigma_t} \right)$$

// regret at t time

this is the regret. And we know that time averaged regret $\leq \varepsilon$.

$$\Rightarrow \frac{1}{T} \sum_{t=1}^{T} \left(\mathop{\mathbb{E}}_{s \sim \sigma_t} [u_i(s_i', s_{-i})] - \mathop{\mathbb{E}}_{s \sim \sigma_t} [u_i(s)] \right) \leq \varepsilon$$

$$\Rightarrow \mathop{\mathbb{E}}_{s \sim \sigma} [u_i(s_i', s_{-i})] - \mathop{\mathbb{E}}_{s \sim \sigma} [u_i(s)] \leq \varepsilon, \quad \left[\text{as } \sigma = \frac{1}{T} \sum_{t=1}^{T} \sigma_t\right]$$

$\Rightarrow \sigma$ is ε-CCE

□

Swap Regret / No Swap Regret

Time averaged swap regret is defined as:

$$\frac{1}{T}\left(\max_{\delta:A\to A}\sum_{t=1}^{T}\sum_{a\in A}p_t(a)\pi_t(\delta(a)) - \sum_{t=1}^{T}\sum_{a\in A}p_t(a)\cdot\pi_t(a)\right)$$

An algorithm is said to have no-swap-regret if its time-averaged swap regret $\to 0$ as $T\to\infty$.

Correlated Equilibrium Revisited:

If every player runs a no-swap regret algorithm (for T iterations) they converge to a CE.

External regret: special case of swap regret ($\delta \in$ set of constant functions).

Theorem 03.10.2024-3: For $\epsilon>0$ suppose after T iterations, time averaged swap regret $\le \epsilon$. Let $\sigma_t = \prod_{i=1}^{n} p_{t,i}$ be the product distribution in t-th iteration for $t\in[T]$ and $\sigma = \frac{1}{T}\sum_{i=1}^{T}\sigma_t$, be the time averaged distribution. Then σ is an ϵ-CE for the game.

Definition (ϵ-CE): σ is an ϵ-CE if $\forall i\in N, \forall \delta_i: S_i\to S_i$

$$\mathbb{E}_{s\sim\sigma}[u_i(s)] \ge \mathbb{E}_{s\sim\sigma}[u_i(\delta_i(s_i), s_{-i})] - \epsilon.$$

Proof of Theorem 03.10.2024-3: Left as Exercise (Straightforward)

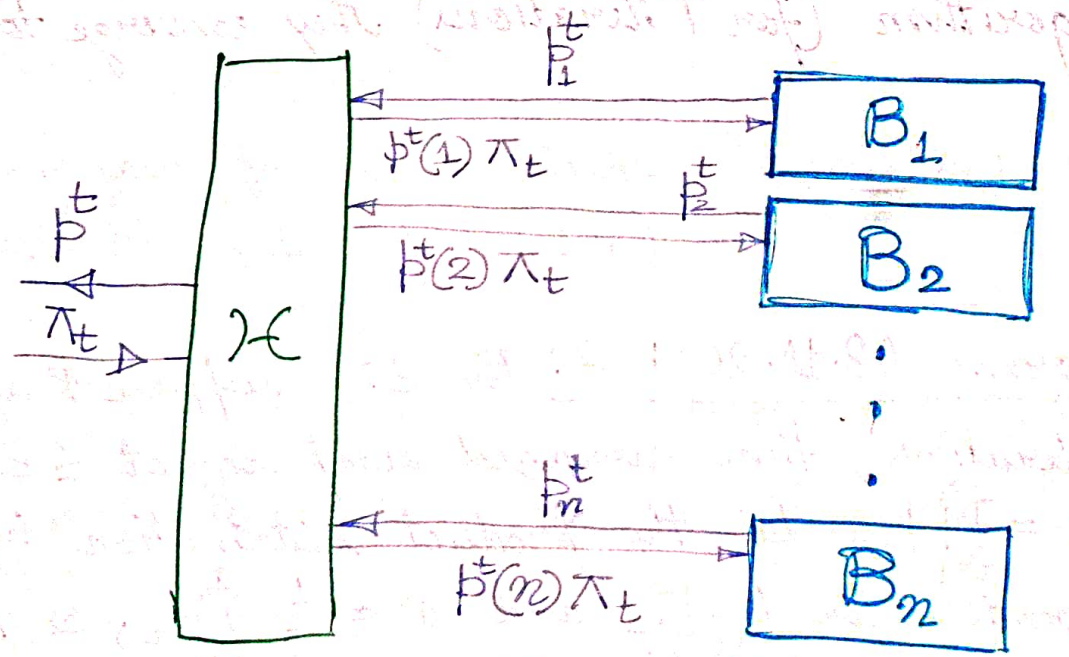
Blackbox reduction from swap regret to external regret

Theorem 18.10.2024-1: Let n be the number of actions ~~with time averaged~~ the swap regret setting. Suppose there exists an algorithm with time averaged external regret $R(T,n)$. Then there is an algorithm with time averaged swap regret $nR(T,n)$.

Proof:

B be a no external regret algorithm, with time averaged external regret $R(T,n)$.

We run n instances of B: $B_1, B_2, \ldots, B_n$, along with a "master algorithm $\mathcal{H}$":



$p^t, p_1^t, \ldots, p_n^t$ are distributions on the set of n actions.

Time averaged ~~external~~ payoff for $H = \frac{1}{T}\sum_{t=1}^{T}\sum_{i\in A} p^t(i)\pi_t(i)$

Time averaged expected payoff under $\delta = \frac{1}{T}\sum_{t=1}^{T}\sum_{i\in A} p^t(i)\pi_t(i)$

For any δ, if we show that the difference of these two quantities goes to 0, then we are done.

Now, B_i has external regret $R(T,n)$. For fixed $\lambda \in A$,

$$\therefore \frac{1}{T}\left(\sum_{t=1}^{T} p^t(i)\pi_t(\lambda) - \sum_{t=1}^{T}\sum_{j\in A} p^t(i)\, p_i^t(j)\pi_t(j)\right) \le R(T,n)$$

This holds true for all $\lambda \in A$, in particular, for $\lambda = \delta(i)$.

$$\therefore \frac{1}{T}\left(\sum_{t=1}^{T} p^t(i)\pi_t(\delta(i)) - \sum_{t=1}^{T}\sum_{j\in A} p^t(i)\, p_i^t(j)\pi_t(j)\right) \le R(T,n).$$

Summing it up for all i, we get

$$\frac{1}{T}\left(\sum_{t=1}^{T}\sum_{i\in A} p^t(i)\pi_t(\delta(i)) - \sum_{t=1}^{T}\sum_{i\in A}\sum_{j\in A} p^t(i)\, p_i^t(j)\pi_t(j)\right) \le nR(T,n)$$

..... (i)

Now, $\sum_{i\in A}\sum_{j\in A} p_i^t(j)\, p^t(i)\pi_t(j) = \sum_{j\in A}\pi_t(j)\sum_{i\in A} p_i^t(j)\, p^t(i)$

$$= \sum_{i\in A}\pi_i(i)\left(\sum_{j\in A} p_j^t(i)\, p^t(j)\right).$$

Therefore, (i) becomes.

$$\frac{1}{T}\left[\sum_{t=1}^{T}\sum_{i\in A} p^t(i)\,\pi_t(\delta(i)) - \sum_{t=1}^{T}\sum_{i\in A} p^t(i)\left(\sum_{j\in A} p_j^t(i)\,p^t(j)\right)\right] \leq n R(T,n)$$

We are done if $\left(\sum_{j\in A} p_j^t(i)\,p^t(j)\right) = p^t(i)$ but at the same time ensuring $\sum_{i\in A} p^t(i) = 1$

However, the problem of finding $p^t(i)$, where

$$\sum_{j\in A} p_j^t(i)\,p^t(j) = p^t(i)$$

$$\sum_{i\in A} p^t(i) = 1$$

is basically same as finding stationary distribution of a finite Markov chain with transition probabilities $p_j^t(i)$; and hence solution always exists.)

This completes the proof. □

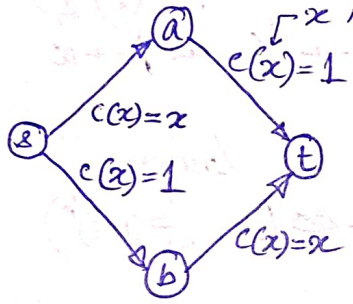
Price of Anarchy

"Helps us quantify the goodness/badness of strategy profiles"

Price of anarchy = ratio of equilibrium utility to the best possible utility for the group of players as a whole.

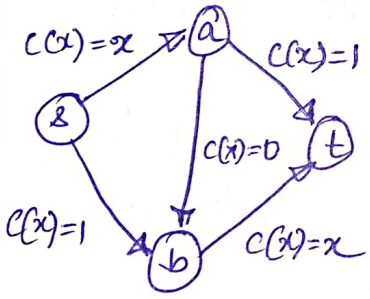
Selfish Routing:

Braess Paradox



Equilibrium: Half traffic through $s \to a \to t$, rest through $s \to b \to t$

Average travel time: $1 + \frac{1}{2} = \frac{3}{2}$.



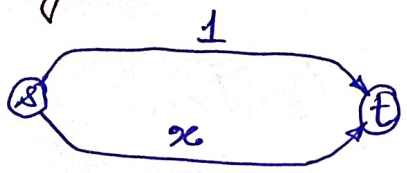
Equilibrium: (SDSE) All traffic goes through $s \to a \to b \to t$

Average travel time: $1 + 1 = 2$

Minimum possible avg. time $= \frac{3}{2}$

Price of Anarchy $= \frac{2}{3/2} = \frac{4}{3}$

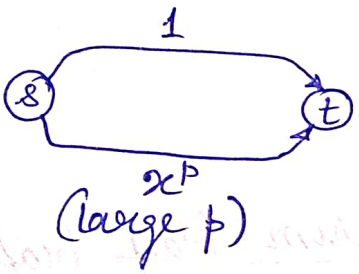
Pigou Network



Equilibrium: All traffic goes through lower-edge
Average travel time: 1

Best possible profile: Half traffic routed through the lower edge; rest through the upper edge. Avg. travel time $= 3/4$

Price of Anarchy (POA) $= \frac{4}{3}$.



Route half traffic, then avg travel time $= \frac{1}{2} + \left(\frac{1}{2}\right)^p \left(\frac{1}{2}\right)$
avg travel time $\to 1/2$ as $p \to \infty$

Route $(1-\varepsilon)$ fraction of traffic on lower edge, then avg. travel time $\to 0$. $(\approx (1-\varepsilon)^p)$
POA can be unbounded.

We are ~~interesting~~ interested in the question of when ~~cost funct~~ POA is close to 1.

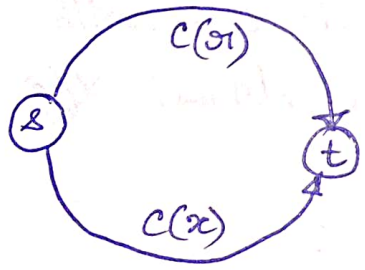
- Highly non-linear $\longrightarrow$ POA not close to 1
- Not too non-linear $\longrightarrow$ POA quite close to 1

Selfish Routing Model:

$G=(V,E)$ directed graph (treated as flow networks)
s: source vertex t: sink vertex
r units of traffic sent from s to t.
Every edge e has a non-negative, non-decreasing, continuous cost function.

We will see that among all networks with cost functions in a set C, the largest is achieved in a Pigou-like network. This means that POA is hammered up due to the non-linearity of the cost functions, and not due to the complexity of the network.

Pigou-like Network:



traffic rate: $r > 0$
x: fraction of traffic through lower edge (units of traffic)
free parameters: c, r

Equilibrium: All traffic routed through edge.
Total travel time: $r \cdot c(r)$.
Minimum possible travel time: $\inf_{0 \le x \le r} \{ x \cdot c(x) + (r-x) \cdot c(r) \}$

$$POA = \sup_{0 \le x \le r} \left\{ \frac{r \cdot c(r)}{x \cdot c(x) + (r-x) \cdot c(r)} \right\}$$

$\mathcal{C}$ be an arbitrary set of non-negative, non-decreasing, continuous cost functions. The Pigou-Bound, $\alpha(\mathcal{C})$ is defined as the worst POA in a Pigou-like networks with cost functions in $\mathcal{C}$.

$$\alpha(\mathcal{C}) = \sup_{c \in \mathcal{C}} \sup_{r \geq 0} \sup_{0 \leq x \leq r} \left\{ \frac{r \cdot c(r)}{x c(x) + (r-x) c(r)} \right\}$$

For class $\mathcal{C} = \{ax+b \mid a,b \geq 0\}$, the set of all affine functions, $\alpha(\mathcal{C}) = \frac{4}{3}$.

Theorem 24.10.2024-1: For every set $\mathcal{C}$ of non-decreasing, non-negative cost functions, and every selfish routing network with cost functions in $\mathcal{C}$, POA is atmost $\alpha(\mathcal{C})$

Turning into flow network:

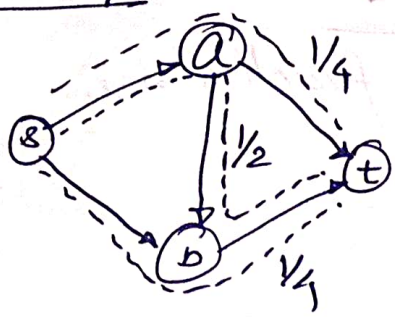
$G=(V,E)$: selfish routing network with r units of traffic travelling from s to t.

$P (\neq \phi)$: set of paths from s to t.

Flow : traffic split over the s-t paths, non negative vector $\{f_p\}_{p \in P}$ with $\sum_{p \in P} f_p = r$

For $e \in E$, and flow f, $f_e = \sum_{\substack{p \in P \\ e \in p}} f_p$.

Example:



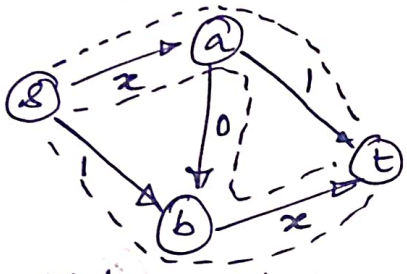
$f_{(a,t)} = f_{(s,b)} = \frac{1}{4}$.

$f_{(s,a)} = f_{(b,t)} = \frac{3}{4}$

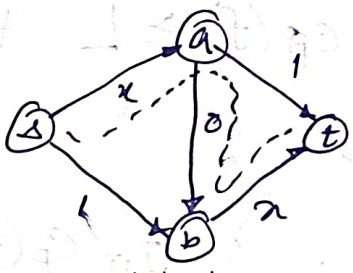
$f_{(a,b)} = \frac{1}{2}$

A flow is an <u>equilibrium</u> iff traffic flows only on shortest s-t paths ("shortest" is defined on the basis of $\{c_e(f_e)\}$), i.e. $f_{\hat{P}} > 0$ iff $\hat{P} \in \text{argmin}\left\{\sum_{e \in P} c_e(f_e)\right\}$

Example:



Not equilibrium



equilibrium

$s \to a \to b \to t$ is the only shortest path.

"In every selfish routing game, $\exists$ at least one equilibrium flow"

Prove that this is equivalent to potential games having at least one PSNE's.

The cost is total travel time: $C(f)$ for flow f.

$$C(f) = \sum_{P \in \mathcal{P}} f_P \cdot c_P(f). \qquad \left[c_P(f_e) = \sum_{e \in P} c_e(f_e)\right.$$

$$= \sum_{e \in E} f_e \cdot c_e(f_e)$$

All equilibrium flows have the same cost.

Proof of Theorem 24.10.2024-1:

Fix network G with cost function in $\mathcal{C}$.

f: equilibrium flow

f^*: optimal (mincost) flow.

$$\therefore f_{\hat{P}} > 0 \Rightarrow c_{\hat{P}}(f) \leq c_P(f) \quad \forall P \in \mathcal{P}$$

(by definition of f).

All paths $\hat{P}$ used by f have a common cost $c_{\hat{P}}(f) = L$ (the min-cost path).

$$c_P(f) \geq L \quad \forall P \in \mathcal{P}.$$

Now,

$$\sum_{P \in \mathcal{P}} f_P \, c_P(f) = r L \quad \text{..... (i)}$$

Again,

$$\sum_{P \in \mathcal{P}} f_P^* \, c_P(f) \geq r L \quad \text{..... (ii)}$$

Rewrite (i) & (ii) as sum of edges and subtract (i) from (ii).

$$\sum_{e \in E} (f_e^* - f_e) \cdot c_e(f_e) \geq 0 \quad \text{..... (iii)}$$

Now,

$$\alpha(\mathcal{C}) = \sup_{c \in \mathcal{C}} \sup_{r \geq 0} \sup_{x \geq 0} \left\{ \frac{r \cdot c(r)}{x \cdot c(x) + (r - x) \cdot c(r)} \right\}$$

$$\therefore \alpha(\mathcal{C}) \geq \frac{\sum_e f_e c_e(f_e)}{f_e^* c_e(f_e^*) + (f_e - f_e^*) c_e(f_e)}, \quad \left[\begin{array}{l}\text{using} \\ c_e \leftarrow c \\ f_e \leftarrow x \\ f_e^* \leftarrow x^*\end{array}\right]$$

Rearranging, we get

$$f_e^* c_e(f_e^*) \geq \frac{f_e c_e(f_e)}{\alpha(\mathcal{C})} + (f_e^* - f_e) c_e(f_e)$$

Summing it for all edges $e \in E$

$$\sum_{e \in E} f_e^* c_e(f_e^*) \geq \frac{1}{\alpha(\mathcal{C})} \sum_{e \in E} f_e c_e(f_e) + \sum_{e \in E} (f_e^* - f_e) c_e(f_e)$$

$$\Rightarrow C(f^*) \geq \frac{C(f)}{\alpha(\mathcal{C})} + \sum_{e \in E} (f_e^* - f_e) c_e(f_e)$$

$$\geq \frac{C(f)}{\alpha(\mathcal{C})} + 0 \quad [\text{from (iii)}]$$

$$\Rightarrow \frac{C(f)}{C(f^*)} = \text{POA} \leq \alpha(\mathcal{C}) \qquad \square$$

Corollary 24.10.2024: For $\mathcal{C} = \{ax+b \mid a, b \geq 0\}$, the set of affine functions, ~~the set of~~ any selfish routing network with cost functions from $\mathcal{C}$, must have POA $\leq \frac{4}{3}$.

Proof: $\alpha(\mathcal{C}) = \frac{4}{3} \Rightarrow \text{POA} \leq \alpha(\mathcal{C}) = \frac{4}{3}$ $\square$

Selfish Load Balancing:

Set of m machines with speeds $s_1, s_2, \ldots, s_m$,
set of n jobs with weight $w_1, w_2, \ldots, w_n$.
We want an assignment of tasks to machines.
$A: [n] \to [m]$, that is as "balanced" as possible.

- No central authority
- Only selfish users aiming to maximize their own individual benefit.

Load of machine j, $l_j = \sum_{\substack{i \in [n] \\ A(i)=j}} \frac{w_i}{s_j}$

Makespan $= \max_{j \in [m]} l_j$

(Selfish) Load Balancing Games:

- n players, player i manages task i, $\forall i \in [n]$
- Set of pure strategies $S_i = [m]$, $\forall i \in [n]$
- Cost of player $i = l_{A(i)}$
- Social cost (for strategy profile A), cost(A) = makesp

Theorem 01.11.2024-1: Every selfish load balancing game has a PSNE.

Proof: Let A be a strategy profile with sorted load vector being $(\lambda_1, \lambda_2, \ldots, \lambda_m)$. Suppose A is not a PSNE. $\exists$ player i s.t. i's task can be moved from machine j to machine k (indices are according to sorted order).

If the sorting was done in a non-increasing order, then $j < k$.

$\therefore$ Load on machine j, l_j decreases, load on machine k will increase, and

increased-final-load-on-machine $k < \lambda_j$.

$\underbrace{\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{j-1}}_{\text{unchanged}}, \lambda_j, \lambda_{j+1}, \ldots, \lambda_{k-1}, \lambda_k, \lambda_{k+1}, \ldots, \lambda_m$

$\lambda'_j < \lambda_j$ $\quad$ $\lambda'_k < \lambda_j$

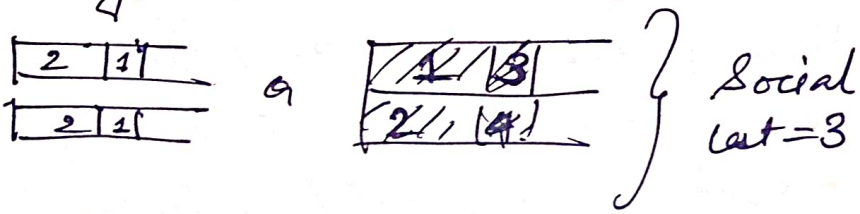
Resulting sorted vector must be lexicographically smaller than the initial vector. This cannot go on forever, as we must reach a feasible vector which is ~~feas~~ the lexigographically smallest. That defines a PSNE.

Example:

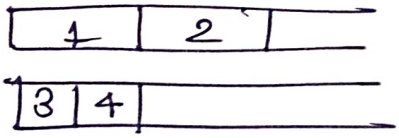
4 task with weights $\overset{1}{2}, \overset{2}{2}, \overset{3}{1}, \overset{4}{1}$

2 identical machines.

Best assignment would be



Consider the assignment:



This is also a PSNE with Social Cost = 4

So we expect price of anarchy to be at least $\frac{3}{4} \cdot \frac{4}{3}$, in selfish load balancing games (with identical machines).

$$POA = \max_{A \in PSNE} \frac{cost(A)}{OPT}$$

Price of Anarchy for selfish load balancing games, with identical machines

Assume $s_1 = s_2 = \dots = s_m = 1$.

Theorem: For any selfish load balancing games with identical machines

$$cost(A) \leq \left(2 - \frac{2}{m+1}\right) OPT.$$

Proof: Let A be any PSNE. Let j^* be a machine with the highest load; Let i^* be the task with the smallest weight assigned to j^*.

Assume that j^* has at least two jobs assigned to it (otherwise $cost(A) = \ell_{j^*} = OPT$; bound trivially holds).

$$\Rightarrow w_{i^*} \leq \frac{\ell_{j^*}}{2} = \frac{cost(A)}{2}.$$

Let $j \in [m] \setminus \{j^*\}$. Notice that, since A is a PSNE, the i^{th} job cannot be moved j^{th} machine and not increase the makespan.

$$\therefore \ell_j \geq \ell_{j^*} - w_{i^*} \geq \frac{cost(A)}{2}, \forall j \in [m] \setminus \{j^*\}$$

$$OPT \geq \frac{\sum_{i \in [n]} w_i}{m} = \frac{\sum_{j \in [m]} \ell_j}{m}$$

$$= \frac{\left(\sum_{j \in [m] \setminus \{j^*\}} \ell_j\right) + \ell_{j^*}}{m}$$

$$\geq \frac{cost(A) + (m-1) \cdot \frac{cost(A)}{2}}{m}$$

$$= \frac{m+1}{2m} cost(A)$$

$$\Rightarrow cost(A) \leq OPT \cdot \frac{2m}{m+1} = \left(2 - \frac{2}{m+1}\right) OPT$$

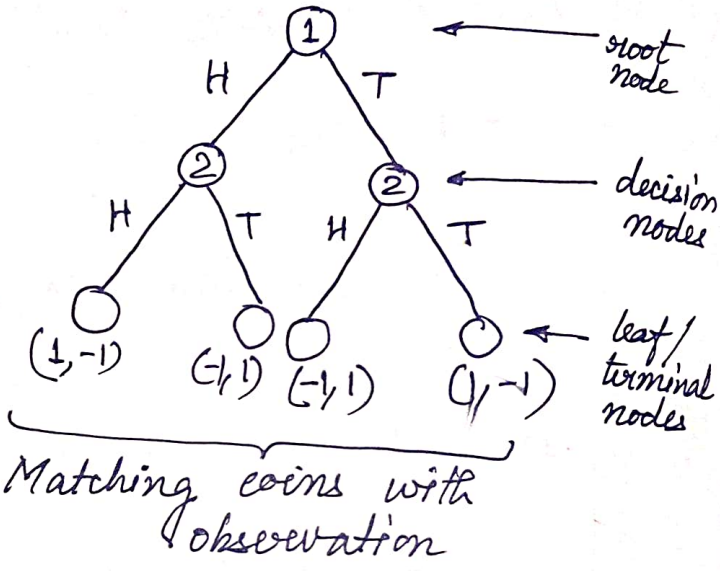
□

Extensive Form Games:

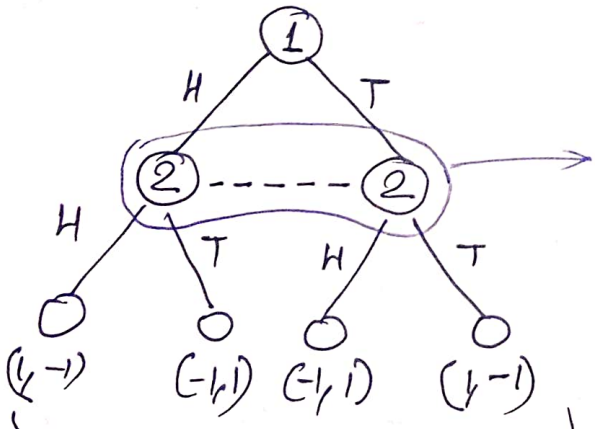
- Captures complete sequential play of a game.

Example: Matching Coins

	H	T
H	(1, -1)	(-1, 1)
T	(-1, 1)	(1, -1)



Matching coins with observation



"Information set" set of decision nodes that are indistinguishable to the player.

matching coins without observation.

Game with perfect information: All information sets are singletons.

An <u>extensive form game</u> is defined as follows

$\Gamma = (N, (S_i)_{i \in N}, \mathcal{H}, P, (u_i)_{i \in N})$

$N = \{1, 2, \ldots, n\}$

$S_i \ (i \in N)$ = set of all available actions for player i

$\mathcal{H}$ = set of all terminal histories (paths from root-to-leaf nodes).

$S_{\mathcal{H}}$ = set of all proper sub-histories (paths from roots to decision nodes.)

$P : S_{\mathcal{H}} \to N$ = mapping each subhistory to a player.

$u_i : \mathcal{H} \to \mathbb{R} \ (i \in N)$ = utility of player i corresponding to each terminal history.

<u>Example</u>: Matching Pennies with observation

$N = \{1, 2\}$

$S_1 = S_2 = \{H, T\}$

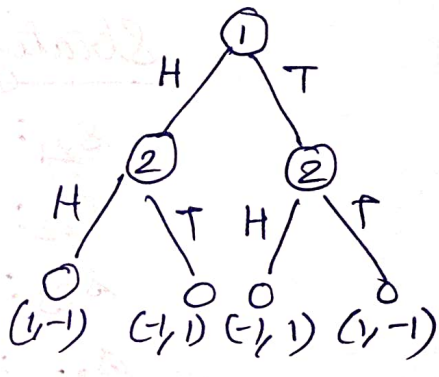
$\mathcal{H} = \{HH, HT, TH, TT\}$

$S_{\mathcal{H}} = \{\varepsilon, H, T\}$

$P(\varepsilon) = 1$

$P(H) = 2$

$P(T) = 2$



	HH	HT	TH	TT
u_1	1	-1	-1	1
u_2	-1	1	1	-1

Let information set of player i be denoted as $\mathbb{I}_i$. For $J \in \mathbb{I}_i$, $C(J)$ denote the set of all possible actions for player i, in the information set j.

A strategy is a mapping $s_i : \mathbb{I}_i \to S_i$ such that $s_i(J) \in C(J) \; \forall J \in \mathbb{I}_i$

Matching coins with observation

$$\mathbb{I}_1 = \{\{\epsilon\}\}$$
$$\mathbb{I}_2 = \{\{H\}, \{T\}\}.$$

Strategies for Player 1

$s_{11} : \{\epsilon\} \to H$
$s_{12} : \{\epsilon\} \to T$

Strategies for player 2

$s_{21} : \{H\} \to H \quad \{T\} \to H$
$s_{22} : \{H\} \to H \quad \{T\} \to T$
$s_{23} : \{H\} \to T \quad \{T\} \to H$
$s_{24} : \{H\} \to T \quad \{T\} \to T$

equivalent normal form game

	s_{21}	s_{22}	s_{23}	s_{24}
s_{11}	(1, -1)	(1, -1)	(-1, 1)	(-1, 1)
s_{12}	(-1, 1)	(1, -1)	(-1, 1)	(1, -1)

Matching coins without observation:

$\Pi_1 = \{\{\varepsilon\}\}$ // typically $\Pi_1 = \{\{P(\varepsilon)\}\}$

$\Pi_2 = \{\{H, T\}\}$ // $\Pi_2 = \{\{P(H), P(T)\}\}$.

Strategies for player 1

$s_{11}: \{\varepsilon\} \to H$

$s_{12}: \{\varepsilon\} \to T$

Strategies for player 2

$s_{21}: \{H, T\} \to H$

$s_{22}: \{H, T\} \to T$

	s_{21}	s_{22}
s_{11}	$(1, -1)$	$(-1, 1)$
s_{12}	$(-1, 1)$	$(1, -1)$

} Equivalent Normal Form Game

We can map any extensive form game to its equivalent normal form game.

$(N, (S_i)_{i \in N}, \mathcal{H}, P, (u_i)_{i \in N})$ exercise $\to$ $(N, (S'_i)_{i \in N}, (u'_i)_{i \in N})$

Bayesian Games:

Private information (about the game) held by each player that others do not know. We call this "type" of the player.

$$\Gamma = (N, (\Theta_i)_{i\in N}, (S_i)_{i\in N}, (p_i)_{i\in N}, (u_i)_{i\in N})$$

N : set of players
Θ_i : set of types (private information) of player i
S_i : set of actions of player i
p_i : distribution $p_i(\cdot|\theta_i)$ over the set Θ_{-i} (denoting the beliefs of player i regarding the type of others)
$u_i : \Theta_i \times S_i \longrightarrow \mathbb{R}$

Then Γ is a Bayesian game:

Strategy: $s_i : \Theta_i \longrightarrow S_i$

Consistency of beliefs:

$(p_i)_{i\in N}$ are consistent if there ~~are~~ is some common probability distribution P over $\Theta = \Theta_1 \times \Theta_2 \times \cdots \times \Theta_n$ if the following holds

Consistency: $$p_i(\theta_{-i}, \theta_i) = \frac{P(\theta_i, \theta_{-i})}{\sum_{t_i \in \Theta_{-i}} P(\theta_i, t_{-i})}$$

$\forall\ \theta_i \in \Theta_i, \forall \theta_{-i} \in \Theta_{-i}, \forall i \in N.$

Example: Two player bargaining game

1 : seller 2 : buyer,

Trade happens only if buyers bid $\geq$ seller's bid.

Bids: $S_1 = S_2 = \{1, 2, \ldots, 10\}$

Types: $\Theta_1 = \Theta_2 = \{1, 2, \ldots, 10\}$

selling price: average of the two bids.

$$p_i(\theta_{-i} \mid \theta_i) = \frac{1}{10}, \quad \forall (\theta_i, \theta_{-i}) \in \Theta$$

$$u_1(\theta_1, \theta_2, s_1, s_2) = \begin{cases} \frac{s_1+s_2}{2} - \theta_1 & \text{if } s_2 \geq s_1 \\ 0 & \text{if } s_2 < s_1 \end{cases}$$

$$u_2(\theta_1, \theta_2, s_1, s_2) = \begin{cases} \theta_2 - \frac{s_1+s_2}{2} & \text{if } s_2 \geq s_1 \\ 0 & \text{if } s_2 < s_1 \end{cases}$$

Now, let $P(\theta_1, \theta_2) = \frac{1}{100}, \forall (\theta_1, \theta_2) \in [10] \times [10]$

$\therefore p_i(\theta_{-i} \mid \theta_i) = \frac{\frac{1}{100}}{\frac{10}{100}} = \frac{1}{10}$, as expected.

$(p_i)_{i \in N}$ is consistent with P.

Sealed bid auctions (first price)

1 seller, 2 bidder/buyers ($N = \{1, 2\}$)

$\Theta_1 = \Theta_2 = [0, \square]$, $S_1 = S_2 = [0, 1]$

$$a_1(s_1, s_2) = \begin{cases} 1 & \text{if } s_1 \geq s_2 \\ 0 & \text{if } s_1 < s_2 \end{cases},$$

$$a_2(s_1, s_2) = \begin{cases} 1 & \text{if } s_2 > s_1 \\ 0 & \text{if } s_2 \leq s_1 \end{cases}$$

Ties are broken by favouring 1.

We assume that $(p_i)_{i \in N}$ follows uniform distribution.

$$p_i([x,y] \mid \theta_i) = y - x.$$

$\forall 0 \le x \le y \le 1, \forall i \in \{1,2\}$

$$u_i(\theta_1, \theta_2, s_1, s_2) = a_i(s_1, s_2) \cdot [s_i - \theta_i]$$

$$= a_i(s_1, s_2) \cdot (\theta_i - s_i)$$

Selten Games

For a Bayesian game $\Gamma = (N, (\Theta_i)_{i \in N}, (S_i), (p_i), (u_i))$

we have a Selten game $\Gamma = (N^s, (S_j^s)_j, (U_j)_j)$

$$N^s = \bigcup_{i \in N} \Theta_i \quad \text{(assuming } \forall i,j \ \Theta_i \cap \Theta_j = \emptyset\text{)}$$

$$S_{\theta_i}^s = S_i$$

$$U_{\theta_i} : \bigtimes_{i \in N} \bigtimes_{\theta_i \in \Theta_i} S_{\theta_i} \longrightarrow \mathbb{R}$$

$$U_{\theta_i}(s_{\theta_i}, s_{-\theta_i}) = \sum_{t_{-i} \in \Theta_{-i}} p_i(t_{-i} \mid \theta_i) \cdot u_i(\theta_i, t_{-i}, s_{\theta_i}, s_{t_{-i}})$$

$$= \mathbb{E}_{p_i(\cdot \mid \theta_i)} \left[u_i(\theta_i, \theta_{-i}, s_{\theta_i}, s_{\theta_{-i}}) \right]$$

Example: Bayesian Pricing Games:

Company 1 produces x_1
Company 2 produces x_2 (similar to x_1) & or product y_2

Prices: $S_1 = \{a_1, b_1\}$
$S_2 = \{a_2, b_2\}$
low high

$\Theta_1 = \{x_1\}$ $\Theta_2 = \{x_2, y_2\}$

$p_2(x_1 | x_2) = 1$ $p_2(x_1 | y_2) = 1$

$p_1(x_2 | x_1) = 0.6$ $p_1(y_2 | x_1) = 0.4$

$\theta_1 = x_1, \theta_2 = x_2$

	a_2	b_2
a_1	(1,2)	(0,1)
b_1	(0,4)	(1,3)

$\theta_1 = x_1, \theta_2 = y_2$

	a_2	b_2
a_1	(1,3)	(0,4)
b_1	(0,1)	(1,2)

Converting to a Selten game:

$N^\Delta = \{x_1, x_2, y_2\}$

$S_{x_1} = \{a_1, b_1\}$, $S_{x_2} = \{a_2, b_2\}$, $S_{y_2} = \{a_2, b_2\}$

$$U_{\theta_i} : S_{x_1} \times S_{x_2} \times S_{y_2} \to \mathbb{R}.$$

$$U_{x_1}(a_1, a_2, a_2) = \sum_{t_{-i} \in \{x_2, y_2\}} p_1(t_{-i} | x_1) \cdot u_1(x_1, t_{-i}, a_1, s_{t_{-i}})$$

$$= (0.6) \cdot u_1(x_1, x_2, a_1, a_2) + (0.4) \cdot u_1(x_1, y_2, a_1, a_2)$$

$$= (0.6) \cdot 1 + (0.4) \cdot 1 = 1.$$

Exercise: complete the entire table:

Equilibria:

$s^* = (s_1^*(\cdot), s_2^*(\cdot), \ldots, s_n^*(\cdot))$ form Bayesian Nash Equilibrium, if $\forall s_i \in S_i, \forall \theta_i \in \Theta_i, \forall \theta_i \in \Theta_i, \forall i \in N$

$$u_i(s_i^*(\theta_i), s_{-i}^*(\theta_i)) \geq u_i(s_i(\theta_i), s_{-i}^*(\theta_i))$$

OR

NE for Bayesian Games	$\equiv$	PSNE for Selten Games

For example, verify that (a_1, a_2, b_2) form a PSNE

BNE: Bayesian Nash Equilibrium.

Example: First price sealed bid auction

θ_1, θ_2 : Independent & uniformly distributed over $[0,1]$

Sealed bid of buyer i takes the form $s_i(\theta_i) = \alpha_i \theta_i$.

Buyer 1:

$\Pr[s_2(\theta_2) \le s_1] \cdot (\theta_1 - s_1)$ is the expected payoff.

$\therefore$ **Objective:** $\max_{s_1 \ge 0} \Pr[s_2(\theta_2) \le s_1] \cdot (\theta_1 - s_1)$

Now, $\Pr[s_2(\theta_2) \le s_1]$

$= \Pr[\alpha_2 \theta_2 \le s_1]$

$= \Pr\left[\theta_2 \le \frac{s_1}{\alpha_2}\right] = \frac{s_1}{\alpha_2}$, $\left[\text{as } s_2(\theta_2) \le \theta_2 \alpha_2 \le \alpha_2 \text{ and } s_1 \le s_2 \le \alpha_2\right]$

$\therefore$ **Objective is:** $\max_{s_1 \in [0, \alpha_2]} (\theta_1 - s_1)\frac{s_1}{\alpha_2}$

Solution:

(PSNE)

(BNE)

$$s_1(\theta_1) = \begin{cases} \frac{\theta_1}{2} & \text{if } \frac{\theta_1}{2} \le \alpha_2 \\ \alpha_2 & \text{if } \frac{\theta_1}{2} \ge \alpha_2 \end{cases}$$

$$s_2(\theta_2) = \begin{cases} \frac{\theta_2}{2} & \text{if } \frac{\theta_2}{2} \le \alpha_1 \\ \alpha_1 & \text{if } \frac{\theta_1}{2} \ge \alpha_1 \end{cases}$$

[if $\alpha_1, \alpha_2 = 1/2$, $\forall \theta_1, \theta_2$, $s_1(\theta_1) = \frac{\theta_1}{2}$, $s_2(\theta_2) = \frac{\theta_2}{2}$]. This is PSNE