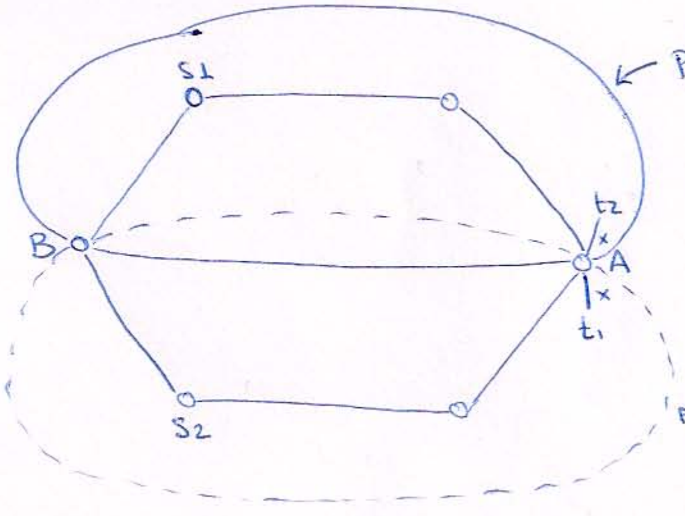
Ονομα: George Kyriakou.

ISP routing problem → ISP competition ⇒ prisoner's dilemma
→ ISPs cooperate



part of the network ISP1 controls

part of the network ISP2 controls

goals:

ISP1 wants to make a transfer from s_1 to t_1
ISP2 wants to make a transfer from s_2 to t_2

↳ tranfer cost from A to t_1 and t_2 is considered ∅

Strategy: will they complete the transfer through point A or point B?

The cost of a transfer is equal to the required number of hops.

Each ISP wants to create the minimum possible traffic within his network. As a result, ISP1 wants to transfer through B (cost 1 for his network) and ISP2 wants to transfer through node A for the same reasons.

social optimum

	A	B
A	(2,2)	(5,1)
B	(5,1)	(4,4)

→ N.E. (each ISP causes cost 1 to himself and 3 to the other)

Obviously PoA = 2

▷ Exercise

N ice-cream sellers must be placed along a beach of length 1: $\underset{0}{\vdash}\!\!\rule[0.5ex]{6em}{0.4pt}\!\!\underset{1}{\dashv}$

The bigger the distance from his neighbors, the greater the profit.

$x_i \in [0,1]$: each ice-cream seller's placement

utility function: $u_i(x_i) = \dfrac{x_{i+1} - x_{i-1}}{2}$ } where x_{i+1} and x_{i-1} are his closest neighbors to his right and left respectively

$u_i(x_i) = \frac{1}{2}\left[(x_{i+1} - x_i) + (x_i - x_{i-1})\right]$

$x_0 = \varnothing$ & $x_{N+1} = 1$ are the ice-cream sellers sitting on $\varnothing$ and 1 respectively and cannot be moved

Can you prove that:

i) if $N=2$: $(x_1^*, x_2^*) = (1/2, 1/2)$ and $\begin{cases} u_1(x_1) = \frac{x_2}{2} \\ u_2(x_2) = \frac{1-x_1}{2} \end{cases}$

ii) if $N=3$: $\nexists$ Nash Equilibrium

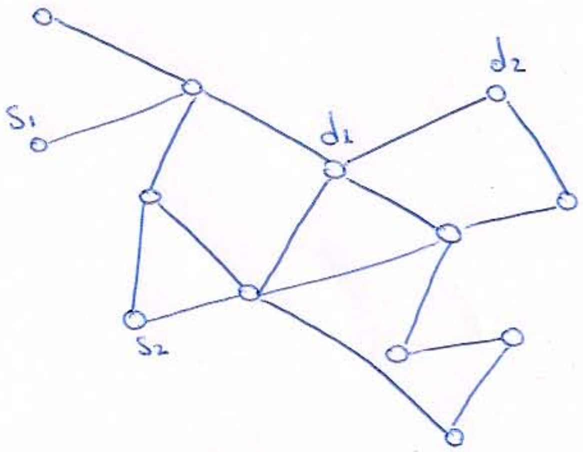
iii) if the ice-cream sellers were placed on a circle, the Nash Equilibrium would be the state when they had equal distances between them

Second Part

- Routing Games
- Congestion Games
- Scheduling Games

▶ Routing Games

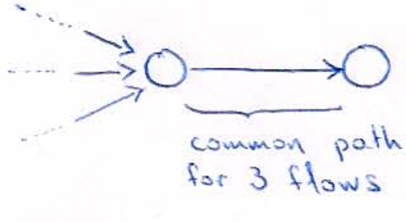
Given a network:



Each player wants to transfer his information from his source (s_i) to his destination (d_i).

When there are common links in the paths

e.g.



all flows/players affect each other.

- There is a load dependent cost (i.e latency) function at each link.

As a result, the more traffic on a link, the longer the delay.

There are two types of links:

i) Splittable (non-atomic): each link can be used by more than one flow at the same time

ii) Unsplittable (atomic): each link can be used only by one flow at a time

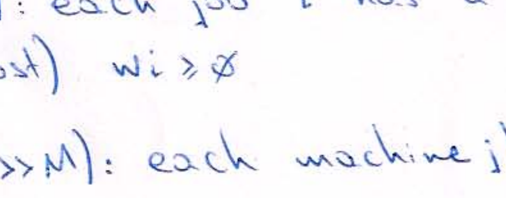
Example

N jobs (players): each job i has a weight (i.e. computational cost) $w_i \geq \emptyset$

M machines ($N \gg M$): each machine j has computational power $s_j \geq \emptyset$

By $a_i = j$ we mean that job i is assigned to machine j.

Let's assume the following jobs assignment:



If L_j is the load at the machine M_j:

$$L_j(\underline{a}) = \frac{\sum_{i:a_i=j} w_i}{s_j}$$

$\sum_{i:a_i=j} w_i$ → the sum of weights of all jobs assigned to machine M_j

s_j → computational power of machine M_j

Vector $\underline{a}$ describes the way jobs are assigned to machines. It can have various forms:

e.g.

i) $\underline{a} = (a_1, a_2, \ldots, a_N)$

↳ its value represents the machine that this job is assigned to

ii)

$$a = \overbrace{\begin{bmatrix} \cdots 010 \cdots \\ \vdots \\ 10 \cdots \\ \vdots \end{bmatrix}}^{M} \Bigg\} N$$

If $a_{ij} = 1$, job i is assigned to machine j.

Since each job can be assigned to only one machine (ie. jobs can't be split) each line can have only one '1'

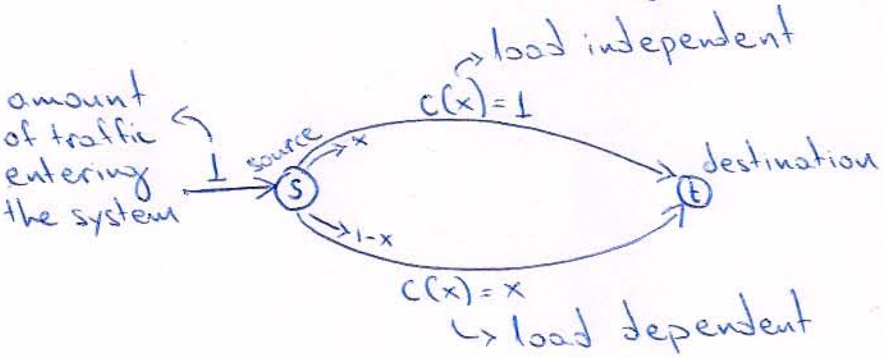
Our goal is to minimize the overall load:

$$\text{minimize} \sum_{j=1}^{M} L_j(\underline{a}) = \text{minimize} \sum_{j=1}^{M} \left(\frac{1}{s_j} \cdot \sum_{i:a_i=j} w_i \right)$$

practically $\frac{w_i}{s_j}$ represents the time machine j needs to complete job i

Example (Pigou 1920)

Given the following network with splittabe links:



→ the traffic entering the system (here is 1) is split into two flows: "x" and "1-x"

→ each link has a different cost function c(x) (representing cost per traffic unit)

global optimum: minimize $x \cdot 1 + (1-x) \cdot (1-x)$ ①

$x \cdot 1 + (1-x)^2 = x + 1 - 2x + x^2 = x^2 - x + 1$

$(x^2 - x + 1)' = 2x - 1$

$2x - 1 = 0 \Rightarrow x^* = \frac{1}{2}$ ②

As a result, the delay in the social optimum is (from ①, ②): $\frac{1}{2} \cdot 1 + (1 - \frac{1}{2})(1 - \frac{1}{2}) \Rightarrow D^*_{s.o.} = 3/4$

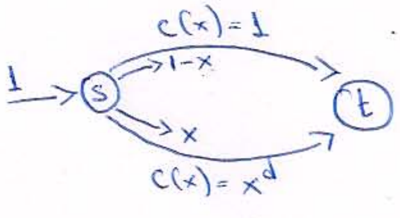
If everyone chose selfish, all players would choose the bottom link. This would cause a delay:

$\emptyset \cdot 1 + 1 \cdot 1 = 1$, since in the Nash Equilibrium $x_{N.E.} = \emptyset$

Price of Anarchy $= \dfrac{\text{delay in N.E.}}{\text{delay in S.O.}} = \dfrac{1}{3/4} = \dfrac{4}{3}$

(*) If the cost function is linearly depented on the load of the link, the Price of Anarchy will not exceed $4/3$, no matter the size of the network, the number of players, latency, etc.

If we have the same problem but without a linear cost function:



global optimum: $\underset{x}{\text{minimize}}\ (1-x) \cdot 1 + x \cdot x^d = 1 - x + x^{d+1}$ (1)

$\dfrac{d(1-x+x^{d+1})}{dx} = \emptyset \Rightarrow x^*(d) = \dfrac{1}{(d+1)^{1/d}}$ (2)

Nash Equilibrium: $x_{N.E.} = 1 \Rightarrow \text{delay} = 0 \cdot 1 + 1 \cdot 1^d = 1$

$$\lim_{d \to +\infty} x^*(d) = \lim_{d \to +\infty} \frac{1}{(d+1)^{1/d}} \overset{a = e^{\log a}}{=\!=\!=} \lim_{d \to +\infty} \frac{1}{e^{1/d \cdot \log(d+1)}} \quad (3)$$

but $$\frac{\log(d+1)}{d} \xrightarrow[d \to +\infty]{} \emptyset \quad (4)$$

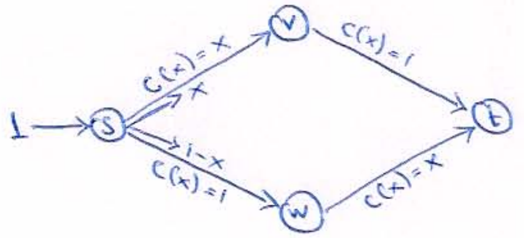
$$\overset{(3)}{\underset{(4)}{\Longrightarrow}} \lim_{d \to +\infty} x^*(d) = \frac{1}{e^0} = 1 \quad (5)$$

$$\overset{(1)}{\underset{(2)}{\Longrightarrow}} 1 - x + \frac{1}{(d+1)^{\frac{d+1}{d}}} \xrightarrow[d \to +\infty]{(5)} 1 - 1 + 0 = \emptyset$$

delay at S.O.

So, $$PoA = \frac{\text{delay at N.C.}}{\text{delay at S.O.}} = \frac{1}{0} \longrightarrow +\infty$$

Braess Paradox



Social Optimum: minimize $x \cdot (x+1) + (1-x)(1+1-x) =$

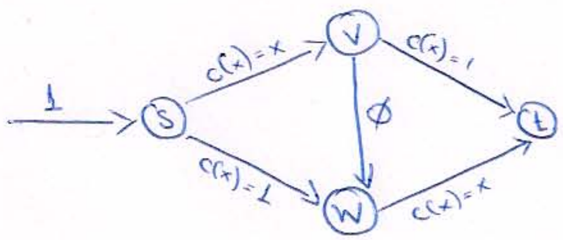
$= x^2 + x + 1 + 1 - x - x - x + x^2 = 2x^2 - 2x + 2$

$$\frac{d(2x^2 - 2x + 2)}{dx} = \emptyset \Rightarrow 4x - 2 = 0 \Rightarrow x^* = 1/2$$

$$\text{delay}_{S.O.} = 2\left(\tfrac{1}{2}\right)^2 - 2\cdot\left(\tfrac{1}{2}\right) + 2 = \frac{3}{2}$$

Nash Equilibrium: $x_{N.E.} = \frac{1}{2} = x^* \Rightarrow PoA = 1$

However, if we add a link 'v → w' with null cost:



we now have three paths with the following costs:

$s \to v \to t: \; x + 1$
$s \to w \to t: \; x + 1$
$s \to v \to w \to t: \; 2x$

since $x \leq 1 \Rightarrow 2x \leq x + 1$ all selfish players in N.E. will choose the third path

As a result, $\text{delay}_{N.E.} = 1\cdot 1 + 0\cdot 1 + 1\cdot 1 = 2$
(Social Optimum will remain the same)

$PoA = \dfrac{2}{3/2} = \dfrac{4}{3}$

Graphs

$G = (V, E)$

G ↓ graph, V ↓ vertex, E ↘ edges

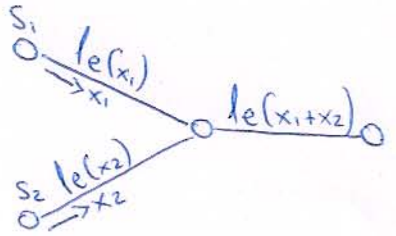
k players want to transmit from source (s_i) to destination (t_i) over a given network (graph)

each player i has a demand function r_i

$$\forall e \in E : \; \underbrace{l_e(\underbrace{x}_{\text{load}}) : R^+ \to R^+}$$

latency function of an edge, depends on load

$\exists$ derivative, non-decreasing



P_i : group of paths that player i uses

$P = \bigcup_i P_i$

f_p : flow over a path p

f_e : flows over an edge e

$$f_e = \sum_{p : e \in p} f_p$$

$$r_i = \sum_{p \in P_i} f_p \Rightarrow \sum_{p \in P_i} \frac{f_p}{r_i} = 1$$

latency of a path: $l_p(f) = \sum_{e \in p} l_e(f_e)$

Overall Cost: $C(f) = \sum_{p\in P} f_p \cdot l_p(f_p) = \sum_{e\in E} \underbrace{f_e}_{\text{traffic}} \cdot \underbrace{l_e(f_e)}_{\text{cost per traffic unit}}$

For a single player i: $C_i(f) = \sum_{p\in P_i} f_p \cdot l_p(f_p) = \sum_{e\in P : p\in P_i} f_e \cdot l_e(f_e)$

Strategy: How will a player split his flow

i) Atomic Routing (unsplittable): $\{P_1, P_2, \dots, P_{|P_i|}\}$

ii) Non-atomic Routing (splittable): $\{f_{P_1}, f_{P_2}, \dots, f_{|P_i|}\}$