Selfish Routing

Thursday, October 27, 2022 11:01 AM

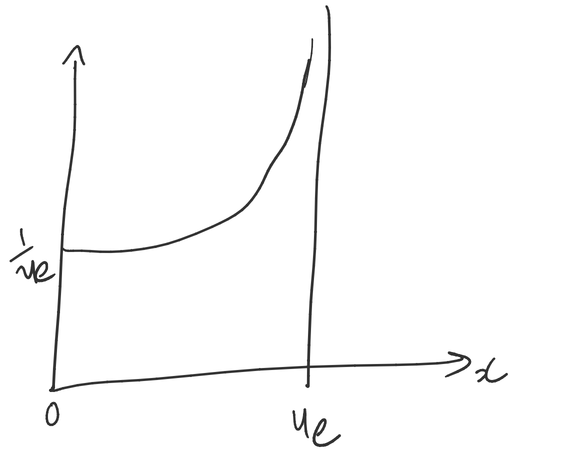
Case Study: N/W over Provisioning.

ISPs: Provide "more capacity than needed"!

- ↳ Improves the overall delay.
- ↳ Better than design + implementation of smart QoS.

$c_e(x) = \dfrac{1}{u_e - x}$ if $x < u_e$ (u_e: capacity)

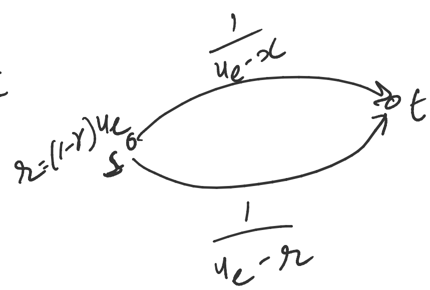
$= \infty$ if $x \geq u_e$



★ n/w is γ-over provisioned if at "NE"

$f_e \leq (1-\gamma) u_e$ (γ fraction of the capacity is not utilized)

$PoA \sim \frac{1}{2}\left(1 + \frac{1}{\sqrt{\gamma}}\right)$ ← exe



$\gamma = 1 \Rightarrow PoA = 1$

$\gamma \to 0 \Rightarrow PoA \to \infty$

$\gamma = 0.1 \Rightarrow PoA \sim 2$!

Thm: Given ℓ, n/w $G = (V, E)$, $s, t \in V$, $r \geq 0$, r units of flow from s to t,

$r \geq 0$, r units of flow.

$\forall e \in E$, $c_e \in \mathcal{C}$

f: NE flow of r units $\equiv$ (ISP example: NE flow of capacity u_e)

f^*: opt flow of $(2r)$ units $\equiv$ (opt " " " $\frac{u_e}{2}$)

Then

$$\text{cost}(f) \leq \text{cost}(f^*)$$

Pf: $L = \min_{P \in \mathcal{P}} c_P(f)$

f is NE $\Rightarrow$ $f_P > 0 \Rightarrow c_P(f) = L$. $\forall P \in \mathcal{P}$.

$$\therefore \text{cost}(f) = \boxed{\sum_{P \in \mathcal{P}} f_P\, c_P(f) = L \sum_{P \in \mathcal{P}} f_P = L \cdot r.}$$

$$\text{cost}\left(\begin{matrix} f^* \text{ w/ cost on each} \\ \text{edge } e \text{ fixed to } c_e(f_e) \end{matrix} \right) = \sum_{P \in \mathcal{P}} f_P^*\, c_P(f) \geq L \cdot \sum_{P \in \mathcal{P}} f_P^*$$

$$= L \cdot (2r)$$

$$\text{cost}(f) = L \cdot r = 2Lr - Lr \leq \sum_{P \in \mathcal{P}} f_P^*\, c_P(f) - Lr$$

$$\overset{?}{\leq} \sum_{P \in \mathcal{P}} f_P^*\, c_P(f^*) = \text{cost}(f^*)$$

$\Updownarrow$

$$\sum_P f_P^*\, c_P(f) - \sum_P f_P\, c_P(f) \leq \sum_P f_P^*\, c_P(f^*)$$

$\Updownarrow$

$$\sum f_P^*\, c_P(f) - \sum f_P^*\, c_P(f^*) \leq \sum_P f_P\, c_P(f)$$

$$\sum_P f_P^* c_P(f) - \sum_P f_P^* c_P(f^*) \leq \sum_P f_P c_P(f)$$

$$\Updownarrow$$

$$\sum_e f_e^* c_e(f_e) - \sum_e f_e^* c_e(f_e^*) \leq \sum_e f_e c_e(f_e)$$

$$\Uparrow$$

<u>claim</u>: $f_e^* \left(c_e(f_e) - c_e(f_e^*)\right) \leq f_e c_e(f_e) \quad \forall e$

<u>pf</u>: case I: $f_e^* \leq f_e$

$$\Rightarrow f_e^* \left(c_e(f_e) - c_e(f_e^*)\right) \leq f_e c_e(f_e)$$

case II: $f_e^* > f_e \Rightarrow c_e(f_e^*) \geq c_e(f_e)$

$$\Rightarrow \left(c_e(f_e) - c_e(f_e^*)\right) \leq 0 \quad \blacksquare$$

Atomic Selfish Routing

$\rightarrow G = (V, E)$

$\rightarrow N = \{1, \dots, n\}$

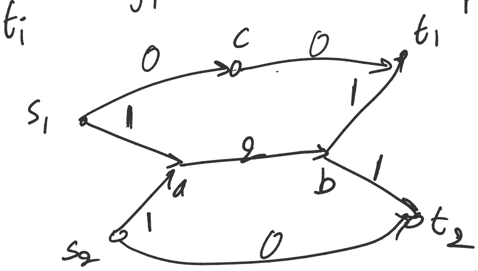
$\rightarrow i \in N$ wants to go from s_i to t_i

$\mathcal{P}_i$ = set of $s_i - t_i$ paths.

$P_i \in \mathcal{P}_i \rightarrow P = (P_1, \dots, P_n)$

$f_e^P = f_e = \#$ agents taking edge e in P.

$\mathcal{P}_1 = \{s_1 - c - t_1, \underbrace{s_1 - a - b - t_1}_{P_1}\}$



$\mathcal{P}_2 = \{s_2 - t_2, \underbrace{s_2 - a - b - t_2}_{P_2}\}$

$\forall e: \quad c_e : \{0, 1, \dots, n\} \rightarrow \mathbb{R}_+$ non-dec.

$$C_i(P) = \sum_{e \in P_i} c_e(f_e^P) \quad \forall i: \quad \underline{\text{cost}(P)} = \sum_{i \in N} C_i(P)$$

$$= \sum \sum c_e(f_e^P)$$

$$C_i(P) = \sum_{e\in P_i} c_e(f_e) \ldots$$

$$= \sum_{i\in N} \sum_{e\in P_i} c_e(f_e^P)$$

$$= \sum_e c_e(f_e) \sum_{\substack{i\in N:\\ e\in P_i}} 1 = \boxed{\sum_e c_e(f_e)\cdot f_e}$$

$$PoA = \max_{NE:f} \frac{cost(f)}{cost(OPT\ flow)}$$

$$\boxed{P \text{ is a NE iff } \forall i:\ C_i(P_i, P_{-i}) \le C_i(q_i, P_{-i}) \quad \forall q_i \in \mathcal{P}_i}$$

$$\mathcal{C} = \{a_e x + b_e \mid a_e, b_e \ge 0\} \ (\text{affine}).$$

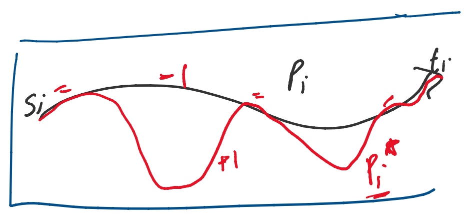
Thm: $PoA(\text{affine}) \le 5/2$ $(\forall e: c_e(x) = a_e x + b_e,\ a_e, b_e \ge 0)$

Proof: P is a NE $\qquad f_e = \#$ agents taking edge e in NE P

P^* is an OPT $\qquad f_e^* =$ " " " OPT P^*

$$P \text{ is NE} \Rightarrow \forall i:\ C_i(P_i, P_{-i}) \le C_i(P_i^*, P_{-i}) \quad \forall i$$

$$= \sum_{e\in P_i^*\setminus P_i} c_e(f_e+1) + \sum_{e\in P_i^*\cap P_i} c_e(f_e)$$



$$\le \sum_{e\in P_i^*} c_e(f_e+1).$$

$$cost(P) = \sum_i C_i(P) \le \sum_i C_i(P_i^*, P_{-i})$$

$$\le \sum \ldots c_e(f_e+1)$$

$$\underline{\text{cost}(P)} = \sum_i c_i(\ldots)$$

$$\leq \sum_i \sum_{e \in P_i^*} c_e(f_e+1)$$

$$= \sum_e c_e(f_e+1) \underbrace{\sum_{\substack{i \in N: \\ e \in P_i^*}} 1}_{= f_e^*}$$

$$= \sum_e c_e(f_e+1)\, f_e^*$$

$$= \sum_{e \in E} \left[a_e(f_e+1) + b_e\right] \cdot f_e^*$$

$$= \sum_{e \in E} a_e \underset{\downarrow\, x}{f_e^*} \underset{\downarrow\, y}{(f_e+1)} + b_e f_e^*$$

$$\left(\begin{array}{c} x, y \in \{0, 1, 2 \ldots\} \\ x(y+1) \leq \frac{5}{3}x^2 + \frac{1}{3}y^2 \end{array} \right) \text{ Magic!}$$

$$\leq \sum_e a_e\left(\frac{5}{3} f_e^{*2} + \frac{1}{3} f_e^2\right) + \frac{5}{3}\sum_e b_e f_e^* + \frac{1}{3}\sum_e b_e f_e$$

$$= \frac{5}{3}\left(\underbrace{\sum_e \underbrace{(a_e f_e^* + b_e)}_{} f_e^*}_{c_e(f_e^*)}\right) + \frac{1}{3}\left(\sum_e \underbrace{(a_e f_e + b_e) f_e}_{c_e(f_e)}\right)$$

$$= \frac{5}{3}\, \text{cost}(P^*) + \frac{1}{3}\, \text{cost}(P).$$

$$\text{cost}(P) \leq \frac{5}{3}\, \text{cost}(P^*) + \frac{1}{3}\, \text{cost}(P)$$

$$\Rightarrow cost(P) \le \frac{5}{3} cost(P^*) + \frac{1}{3} cost(P)$$

$$\Rightarrow \left(1-\frac{1}{3}\right) cost(P) \le \frac{5}{3} cost(P^*)$$

$$\Rightarrow \boxed{PoA = \frac{cost(P)}{cost(P^*)} \le \frac{5/3}{2/3} = \frac{5}{2}}$$

---

0. P: NE
   P^*: OPT.

1. $cost(P) = \sum_{i \in N} C_i(P_i, P_{-i}) \le \sum_{i \in N} C_i(P_i^*, P_{-i})$
   ($\because$ P is a NE).

2. $\sum_{i \in N} C_i(P_i^*, P_{-i}) \le \underbrace{\frac{5}{3}}_{\lambda} cost(P^*) + \underbrace{\frac{1}{3}}_{\mu} cost(P)$.

3. $\textcircled{1}, \textcircled{2} \Rightarrow \left(1-\frac{1}{3}\right) cost(P) \le \frac{5}{3} cost(P^*)$

$$\Rightarrow PoA = \frac{cost(P)}{cost(P^*)} = \frac{\overbrace{5/3}^{\lambda}}{(1-\underbrace{1/3}_{\mu})}$$

★ (λ, μ) - Smooth Game (for $\mu \le 1$).

for any P, P^*

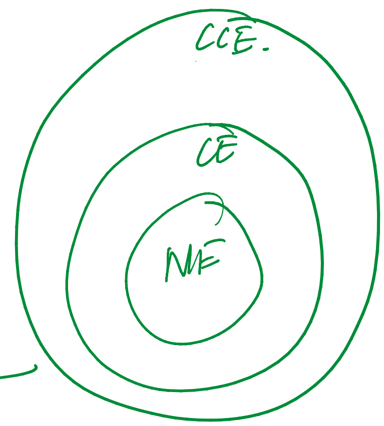
$+ \mu \, cost(P)$

for any p, p^*

$$\sum_{i \in N} C_i(p_i^*, p_{-i}) \le \lambda \, cost(p^*) + \mu \, cost(p)$$

Thm: PoA of (λ, μ)-smooth game is

$$\le \frac{\lambda}{(1-\mu)}$$



$$PoA^{NE} \le PoA^{CE} \le PoA^{CCE} \le \frac{\lambda}{(1-\mu)}$$

$$PoA = \frac{\text{cost at worst eq}}{\text{OPT cost}}.$$