M.Sc. Program in Computer Science
Department of Informatics

Algorithmic Game Theory
Selfish Routing

Vangelis Markakis
markakis@gmail.com

Selfish routing

- In mechanism design, we studied how to enforce a particular strategy (the truthful one)
- We designed the rules of the game so that being truthful was a dominant strategy of the game
- In many other settings, we cannot design a game from scratch
- But we can observe or recommend strategies
- Goal: Evaluate the equilibria of a game, as the outcomes more likely to occur

Non-atomic selfish routing

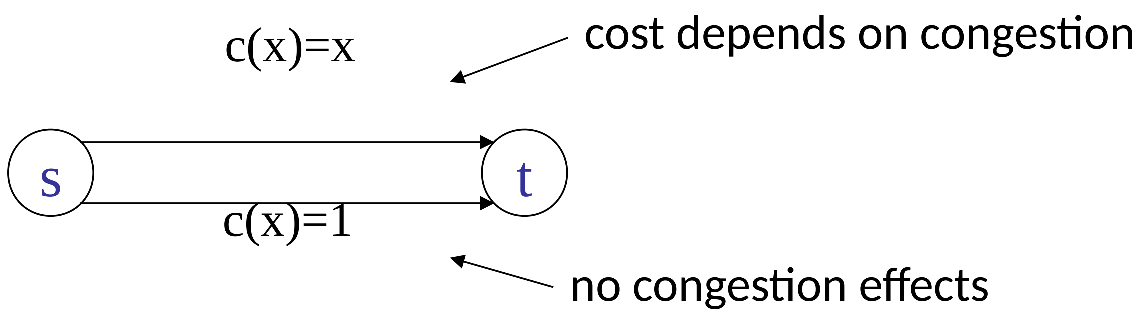
Nonatomic selfish routing

Informal description

- Consider a directed graph depicting a network
- Users want to send traffic from a start point to some end point
- Each user controls an infinitesimally small quantity of traffic
- The traffic needs to cross the edges of a path to reach the destination
- Each edge incurs a cost (time delay, etc)
- The cost depends on the traffic volume crossing the edge

Pigou's Example

[Pigou 1920]: One unit of traffic wants to go from s to t

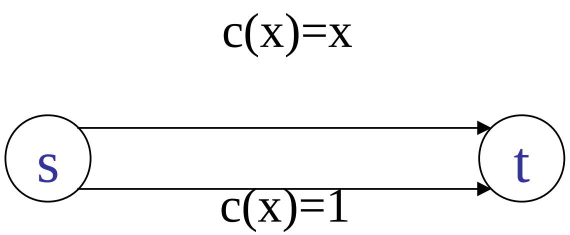


Q: what will selfish network users do?

- assume everyone wants smallest-possible cost

Pigou's Example

Claim: All traffic will take the top link

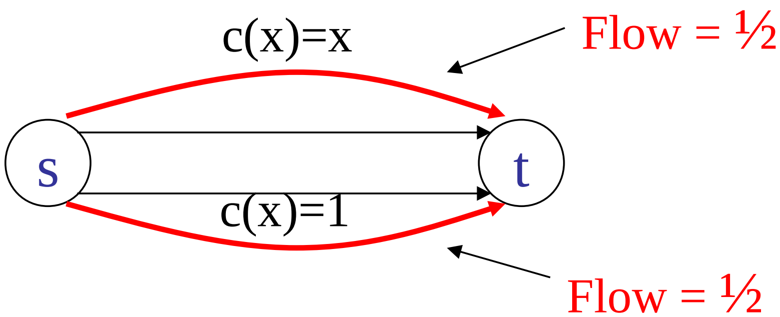


Reason:

- Suppose an ε-fraction of traffic takes the bottom link
- 1-ε on the upper link
- The users on the bottom link are envious
- Only way to have an equilibrium is for everybody to take the top link
- Average delay = 1

Can We Do Better?

- We take the average delay as a metric for the network performance
- Consider instead: traffic split equally

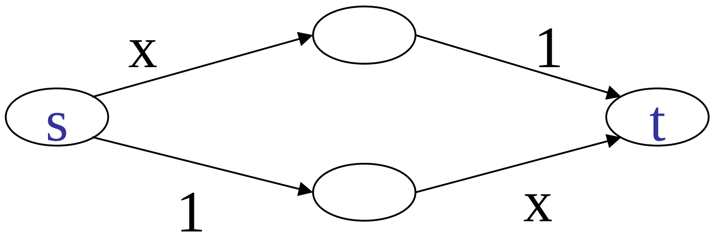


Improvement:

- half of the traffic has cost 1 (same as before)
- half of the traffic has cost ½ (much improved!)
- Average delay: $\frac{1}{2} \cdot 1 + \frac{1}{2} \cdot \frac{1}{2} = \frac{3}{4}$

Braess Paradox

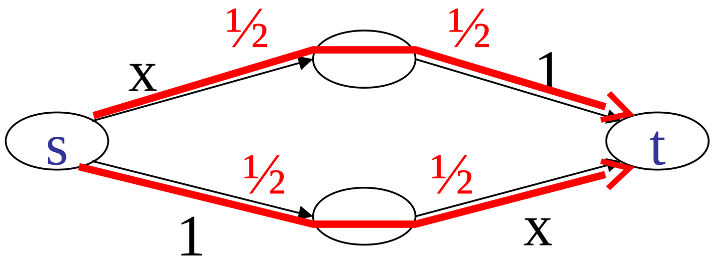
Initial Network:



- Suppose again 1 unit of traffic wants to go from s to t
- Equilibrium flow: equal split
- ½ of the traffic takes the upper route
- The rest take the bottom route
- In any other split some users will have incentives to deviate

Braess Paradox

Initial Network:

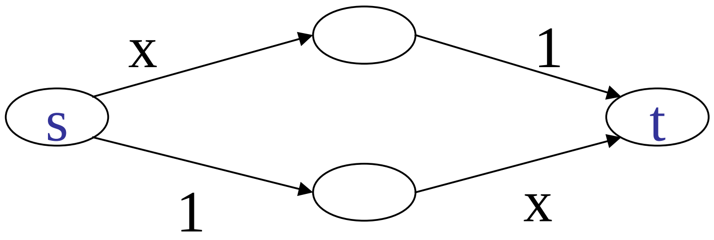


Delay in each route = ½ + 1 = 1.5
Average delay = 3/2

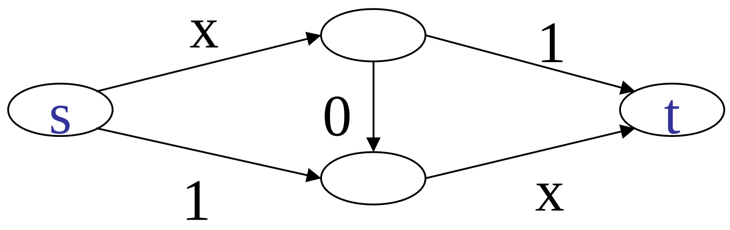
- Suppose the government is thinking of adding 1 very fast new road to help decrease the congestion

Braess Paradox

Initial Network:



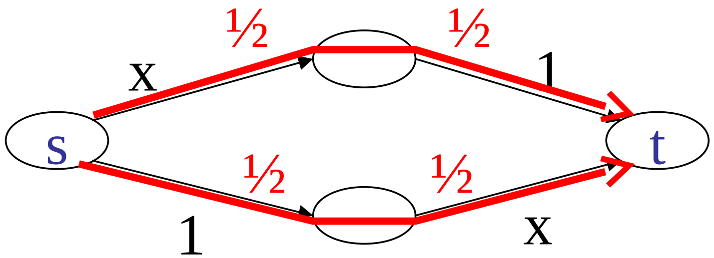
Augmented Network:



- What will the network users do in the augmented network?
- Unique equilibrium to use the route with the fast road

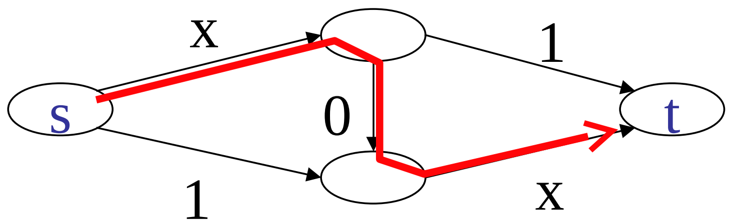
Braess Paradox

Initial Network:



Cost = 1.5

Augmented Network:



Cost = 2

All traffic incurs more cost! [Braess '68]

Selfish Routing Games

Formal description:

- directed graph $G = (V,E)$
- source-destination pairs $(s_1,t_1), \ldots, (s_k,t_k)$
- r_i = amount of traffic that needs to go from s_i to t_i
  - The traffic can be split into different paths from s_i to t_i
- for each edge e, a cost function $c_e()$
  - Assumed continuous, non-negative, and nondecreasing
  - Depends on the traffic crossing edge e
  - Usually expresses the delay of the traffic crossing edge e

Selfish Routing Games

Players

- Each player controls an infinitesimally small amount of flow
  - cars in a road network
  - packets in a network

Outcomes of a selfish routing game: feasible flows

- Need to specify the flow routed on every path connecting some s_i to t_i
- For an s_i-t_i path p, f_P = amount of traffic choosing p

Feasible flow vectors:

- $f_p \geq 0$, for every path p connecting some s_i to t_i
- For i=1,..., k, total flow on all s_i-t_i paths must equal the demand r_i

Selfish Routing Games

Consider a feasible flow f

- f can be written as a vector specifying the flow f_p for every path p connecting some s_i to t_i
- Let P_i = set of all distinct paths from s_i to t_i
- Let $P_{all} = \cup_i P_i$ = all the paths in the graph that are of interest to us
- f has a coordinate f_p for every $p \in P_{all}$

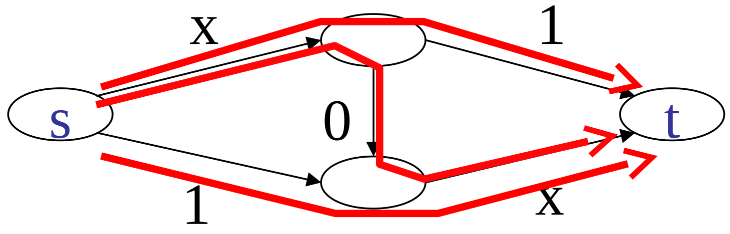
Representation as an edge flow vector:

- We can also write f as a vector along edges of the graph
- For every edge e, $f_e = \Sigma_{p:\, e \in p}\, f_p$
- We need this representation since the delay is evaluated per edge

Selfish Routing Games

Example:

- As a path vector we would need to specify 3 values for the 3 possible paths
- Let
  - p1 be the upper path
  - p2 be the bottom path
  - p3 be the path using the fast link



- A feasible flow for 1.2 units of traffic: f = (0.5, 0.3, 0.4)
- As an edge flow vector:
  - sum in each edge e the flow that goes through e
  - E.g., for the upper rightmost edge: $f_e = 0.9$

Utility functions vs latencies

- To complete the description of the game, we need to define the utility function of a player
- Each player here is choosing a path
- It is more convenient to talk about latency/cost rather than utility
- Given a feasible flow f
  - Latency/cost on an edge e: $c_e(f_e)$ = cost experienced by the traffic going through edge e
  - Latency/cost on a path $p \in P_{all}$: $c_p(f) = \sum_{e \in p} c_e(f_e)$

Equilibrium flows

- When can we say that a flow is at equilibrium?
- When no arbitrarily small quantity of traffic can have an incentive to deviate
- Consider a feasible flow f, and a player controlling a δ amount of flow, who has chosen a path $p_1 \in P_i$
- New flow after a deviation to a path p_2:

$$f' = \begin{cases} f_p - \delta, & \text{if } p = p_1 \\ f_p + \delta, & \text{if } p = p_2 \\ f_p, & \text{o.w.} \end{cases}$$

- Definition: A feasible flow f is a Nash equilibrium flow if for any i = 1, …, k, any $p_1, p_2 \in P_i$, with $f_{p1}>0$, and $\delta \in [0, f_{p1}]$

$$c_{p1}(f) \leq c_{p2}(f')$$

Equilibrium flows

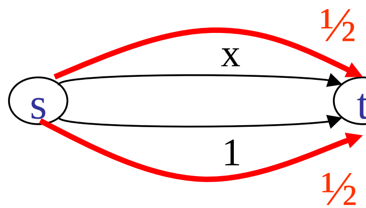
Due to continuity of the cost functions:

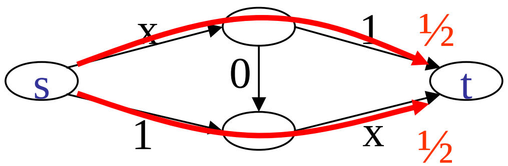
Equivalent definition: [Wardrop '52] A flow f is a Nash flow if for any $i = 1, \ldots, k$, and any $p_1, p_2 \in P_i$, with $f_{p1} > 0$,

$$c_{p1}(f) \leq c_{p2}(f)$$

I.e., all flow is routed on min-cost paths [given current edge congestion]

Examples of non-equilibrium flows:





Existence

- When can we guarantee that a Nash flow exists?
- Lemma: If the cost function of every edge is continuous and non-decreasing, then the game admits a Nash flow with pure strategies
- Existence can be actually guaranteed for a wider class of congestion games (next lecture)
- Main conclusion: no matter how complex the network is, there is a way that the users can reach an equilibrium

Wardrop Equilibrium (Nash flow)

A feasible flow is a Wardrop equilibrium if for every commodity *i* :

$$\forall p, q \in P_i, f_p > 0 : c_p(f) \leq c_q(f)$$

Intuitively, no player has incentive to deviate

Moreover: $\forall p, q \in P_i : f_p > 0, f_q > 0 \Rightarrow c_p(f) = c_q(f)$

Existence and Uniqueness

Let $\Phi(f) := \sum_{e\in E} \int_0^{f_e} c_e(x)dx$

Assume ***f*** is an equilibrium flow.

Change ***f*** to a feasible flow ***f'*** that differs with ***f*** in only two paths (p, q) of the same commodity: $f'_p = f_p - \delta,\ f'_q = f_q + \delta$

$$\Phi(f') - \Phi(f) = \sum_{e\in p\cup q} \int_0^{f'_e} c_e(x)dx - \sum_{e\in p\cup q} \int_0^{f_e} c_e(x)dx$$

$$\Downarrow$$

$$\Phi(f') - \Phi(f) = \sum_{e\in q-p} \int_{f_e}^{f_e+\delta} c_e(x)dx - \sum_{e\in p-q} \int_{f_e-\delta}^{f_e} c_e(x)dx$$

$$\Downarrow$$

$$\text{for } \delta \to 0:$$

$$\Phi(f') - \Phi(f) \approx \sum_{e\in q-p} \delta c_e(f'_e) - \sum_{e\in p-q} \delta c_e(f_e) = \delta(c_q(f') - c_p(f)) \geq 0$$

Existence and Uniqueness

Consider the convex program CP:

$$\min \ \Phi(f) := \sum_{e \in E} \int_0^{f_e} c_e(x)dx$$

so that

$$\sum_{p \in P_i} f_p = r_i, \forall i \in \{1 \ldots k\}$$

$$f_e = \sum_{p \in P : e \in p} f_p, \forall e \in E$$

$$f_p \geq 0, \forall p \in P$$

By Karush-Kuhn-Tucker optimality conditions:

$$\text{A feasible flow f is optimal for CP}$$

$$\Updownarrow$$

$$h'_p := \sum_{e \in p} \left(\int_0^{f_e} c_e(x)dx\right)' \leq \sum_{e \in q} \left(\int_0^{f_e} c_e(x)dx\right)' = h'_q,$$

$$\forall i \in \{1 \ldots k\}, \forall p, q \in P_i, f_p > 0$$

$$\Leftrightarrow \quad c_p(f) \leq c_q(f)$$

Optimal Flow

A feasible flow ***f**** is optimal if for every feasible flow ***x***:

$$C(f^*) \leq C(x) \qquad \left(C(f) = \sum_{e \in E} f_e c_e(f_e)\right)$$

Once again:

$$\begin{array}{l} \min \sum_{e \in E} c_e(f_e) f_e \\ \text{so that} \\ \sum_{p \in P_i} f_p = r_i, \forall i \in \{1 \ldots k\} \\ f_e = \sum_{p \in P : e \in p} f_p, \forall e \in E \\ f_p \geq 0, \forall p \in P \end{array}$$

By KKT conditions

$$f^* \text{ optimal} \Leftrightarrow c_p(f^*) + \sum_{e \in p} c'_e(f^*_e) f^*_e \leq c_q(f^*) + \sum_{e \in q} c'_e(f^*_e) f^*_e,$$

$$\forall i \in \{1 \ldots k\}, \forall p, q \in P_i, f_p > 0$$

Evaluating equilibria

- To evaluate the performance of Nash equilibria, we need to consider the derived social welfare
- Social welfare vs social cost: Since we considered the cost/latency for each user, it is more natural to consider the social cost as our performance measure: i.e., the average delay experienced in the network
- Definition: Given a feasible flow f, the social cost of f is

$$C(f) = \Sigma_p f_p c_p(f) = \Sigma_e f_e c_e(f_e)$$

- Theorem: All the equilibrium flows attain the same social cost
  - Follows again from the fact that cost functions are continuous and non-decreasing

Price of Anarchy in selfish routing

Q: How bad are the equilibria of a selfish routing game?

•Let f^* be an optimal flow (minimizing the social cost) and f be an equilibrium flow

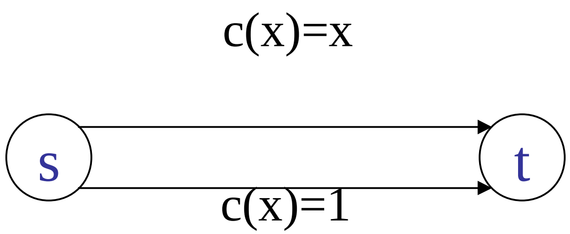
•Given a class of selfish routing games,

$$PoA = \max C(f)/C(f^*)$$

- The maximization is w.r.t. all the games of the class under consideration
- E.g., how bad is PoA for arbitrary cost functions?
- For special classes of cost functions?

Price of Anarchy in selfish routing

- Let's start with linear (affine) cost functions
- Suppose that for every edge e, $c_e(f_e) = a_e f_e + b_e$, for some constants a_e, b_e
- Recall that the examples of Pigou and Braess fall under this class



- Pigou's example shows that PoA ≥ 4/3
- Can it get worse for more complex networks?

How bad is selfish routing?

Theorem [Roughgarden, Tardos '00]: For the class of selfish routing games with a linear cost function on each edge

$$\text{PoA} = 4/3$$

- Independent of the network topology, no matter what the graph looks like!
- Pigou's example achieves the worst-case scenario
- Main take-home message: If the cost functions are linear, selfish behavior cannot affect too much the network performance

How bad is selfish routing?

Main ingredients of the proof for linear cost functions

•Formulate the problem as a convex optimization problem

$$\min \quad C(f) = \sum_{e \in E} f_e \cdot c_e(f_e)$$

s. t.:

$$\sum_{p \in P_i} f_p = r_i, \qquad \forall i \in \{1, 2, \ldots, k\}$$

$$f_e = \sum_{p : e \in p} f_p, \qquad \forall e \in E$$

$$f_p \geq 0 \qquad \forall p \in P_{all}$$

The analysis of the convex program shows that

Lemma: A Nash flow of a given instance is an optimal flow for the instance where traffic is reduced to half ($r_i' = r_i/2$)

How bad is selfish routing?

Main ingredients of the proof for linear cost functions

- Remaining proof relates the optimal flow with the optimal at half the traffic
- Main consequence from the proof: If every cost function is in the form $c_e = a_e f_e$ (no constant term), then PoA = 1!
    - No loss in performance in this case

How bad is selfish routing?

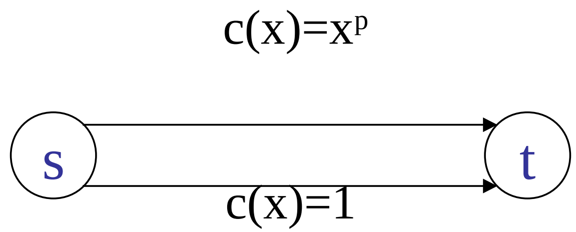
- Generalizing: What about non-linear cost functions?
- It is natural to assume polynomial cost functions as the next step

Description	Typical Representative	Price of Anarchy
Linear	$ax + b$	$4/3$
Quadratic	$ax^2 + bx + c$	$\frac{3\sqrt{3}}{3\sqrt{3}-2} \approx 1.6$
Cubic	$ax^3 + bx^2 + cx + d$	$\frac{4\sqrt[3]{4}}{4\sqrt[3]{4}-3} \approx 1.9$
Quartic	$ax^4 + bx^3 + cx^2 + dx + e$	$\frac{5\sqrt[4]{5}}{5\sqrt[4]{5}-4} \approx 2.2$
Degree $\leq p$	$\sum_{i=0}^{p} a_i x^i$	$\frac{(p+1)\sqrt[p]{p+1}}{(p+1)\sqrt[p]{p+1}-p} \approx \frac{p}{\ln p}$

- PoA can become unbounded as p -> ∞
- But as long as we have low degree polynomials, PoA does not grow too much

How bad is selfish routing?

- Can we understand the worst-case scenarios under non-linear cost functions?
- A non-linear Pigou-like network for polynomial cost functions of degree p:



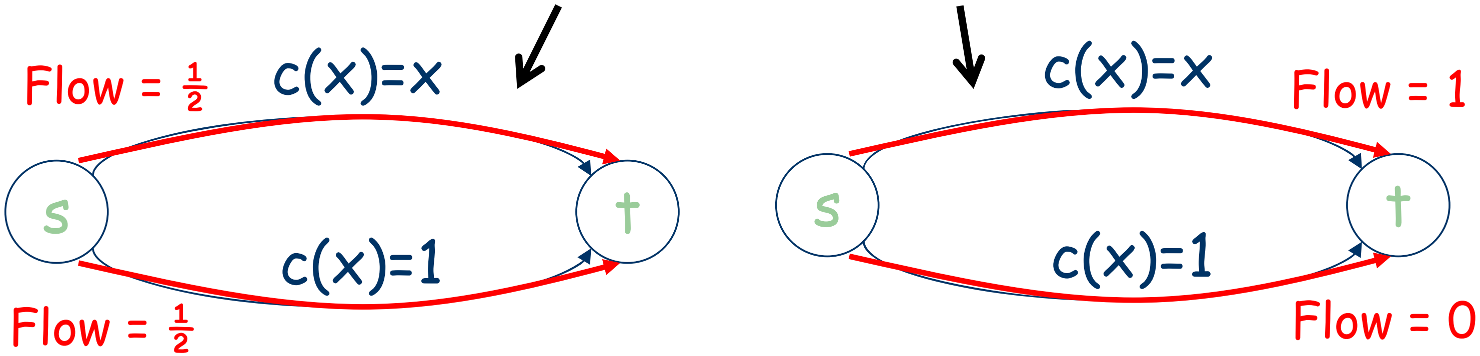
Theorem (informal statement): The worst-case PoA is achieved at Pigou-like networks

Price of Anarchy (PoA)

A measure for the inefficiency of the network:

$\rho(G, r, c) = PoA := \frac{C(f)}{C(f^*)}$, f an equilibrium flow and f^* an optimal flow

Example: Optimal flow (OPT) and Equilirium flow (WE)



$$C(f^*) = (\tfrac{1}{2}) \cdot (\tfrac{1}{2}) + \tfrac{1}{2} \cdot 1 = \tfrac{3}{4},\ C(f) = 1 \text{ and } PoA = \frac{C(f)}{C(f^*)} = \frac{4}{3}$$

Variational Inequality

Variational inequality:

$$f \text{ Wardrop equilibrium} \Leftrightarrow \sum_{e \in E} c_e(f_e) f_e \le \sum_{e \in E} c_e(f_e) f_e^*, \forall f^* \text{ feasible}$$

- The $\Leftarrow$ part: consider *f** differing from *f* in two "same commodity" paths by *δ>0* units (for all commodities).

$$\sum_{e \in E} c_e(f_e) f_e \le \sum_{e \in E} c_e(f_e) f_e^* \Rightarrow \sum_{e \in p} c_e(f_e)\Big(f_e - (f_e - \delta)\Big) \le \sum_{e \in q} c_e(f_e)\Big((f_e + \delta) - f_e\Big)$$

- The $\Rightarrow$ part: same commodity "nonzero" paths are the cheapest of the commodity *i* and cost equal (say $c_i(f)$). Thus

$$\sum_i \sum_{p \in P_i} c_p(f) f_p = \sum_i c_i(f) \sum_{p \in P_i} f_p = \sum_i c_i(f) \sum_{p \in P_i} f_p^* = \sum_i \sum_{p \in P_i} c_i(f) f_p^* \le \sum_{p \in P} c_p(f) f_p^*$$

$$\sum_{p \in P} c_p(f) f_p \le \sum_{p \in P} c_p(f) f_p^* \Rightarrow \sum_{e \in E} c_e(f_e) f_e \le \sum_{e \in E} c_e(f_e) f_e^*$$

Bounding the PoA

Let *f* be an equilibrium flow and *f** an optimal:

$$C(f) = \sum_{e \in E} c_e(f_e) f_e \le \sum_{e \in E} c_e(f_e) f_e^* = \sum_{e \in E} \Big(c_e(f_e) f_e^* + c_e(f_e^*) f_e^* - c_e(f_e^*) f_e^* \Big) \Rightarrow$$

$$C(f) \le \sum_{e \in E} c_e(f_e^*) f_e^* + \sum_{e \in E} \big(c_e(f_e) - c_e(f_e^*) \big) f_e^* = C(f^*) + \sum_{e \in E} \big(c_e(f_e) - c_e(f_e^*) \big) f_e^*$$

We bound the last term:

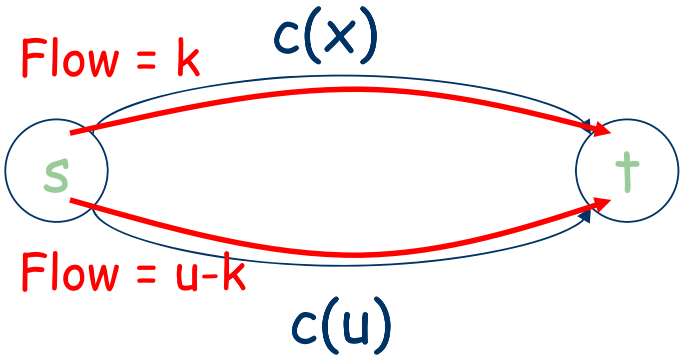
$$f_e^* \Big(c_e(f_e) - c_e(f_e^*) \Big) \le v(f_e, c_e) f_e c_e(f_e), \quad v(u, c_e) = \frac{1}{u c_e(u)} max_{x \ge 0} \{ x(c_e(u) - c_e(x)) \}$$

Let $v(c_e) = \sup_{u \ge 0} v(u, c_e)$ and $v(D) = \sup_{c_e} v(c_e)$ where D is the family of the cost functions. We get

$$\sum_{e \in E} \big(c_e(f_e) - c_e(f_e^*) \big) f_e^* \le v(D) \sum_{e \in E} c_e(f_e) f_e \Rightarrow C(f) \le \frac{1}{1 - v(D)} C(f^*)$$

Tightness

Assume that *u* units are to be routed from *s* to *t*.



At WE everybody goes up

OPT minimizes: $kc(k) + (u-k)c(u)$

$$PoA = \frac{uc(u)}{\min_{k\in[0,v]}[(u-k)c(u)+kc(k)]} = \max_{k\in[0,v]}\left((1-k)+k\frac{c(k)}{uc(u)}\right)^{-1} = \left[1 - \max_{k\in[0,v]} k\left(\frac{c(u)-c(k)}{uc(u)}\right)\right]^{-1}$$

Previous slide: $PoA \leq \left(1 - \sup_{c_e\in D, u\geq 0} \max_{x\geq 0} \frac{\{x(c_e(u)-c_e(x))\}}{uc_e(u)}\right)^{-1}$

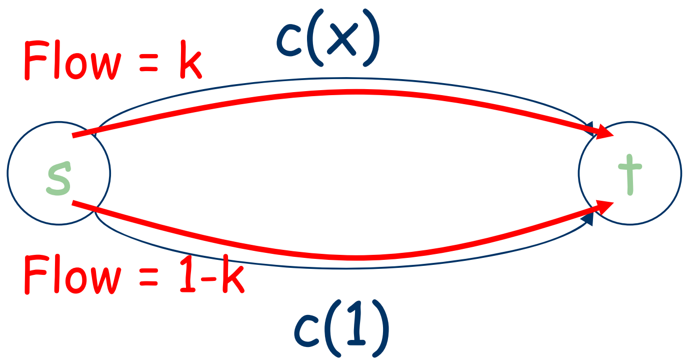
Special cases

- For linear latency functions: $v(D) = \frac{1}{4}$ and $PoA \leq \frac{4}{3}$
- For polynomial of degree *d* latency functions:

$$v(D) = \frac{d}{(d+1)^{(d+1)/d}} \text{ and } PoA \leq \left(1 - \frac{d}{(d+1)^{(d+1)/d}}\right)^{-1}$$

1 unit is to be routed.
At WE everybody goes up
For $c(x) = x^d$ OPT minimizes:

$$k \cdot k^d + (1-k)$$



It is $k = \sqrt[d]{\frac{1}{d+1}}$ and $OPT = 1 - \frac{d}{(d+1)^{d+1/d}}$