Games and Networks and the Quality of Outcomes

Éva Tardos
Cornell University

Plan for the talks

Network Games and Quality of Nash

- Examples of Games in Networks
- Outcome : Nash
- Quality = Price of Anarchy

Learning in Network Games

Quality in other games: Ad-Auctions

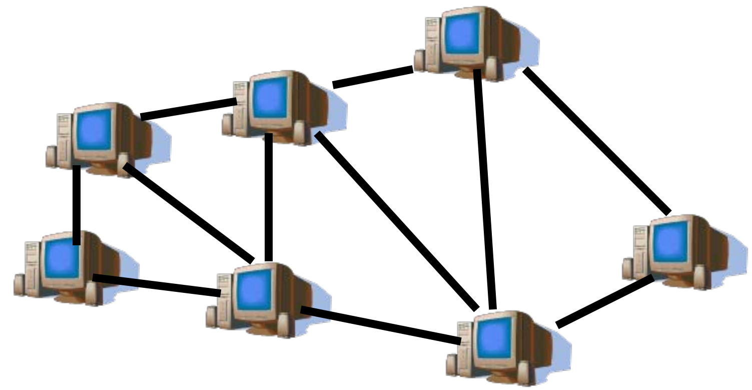
Why care about Games?

Users with a multitude of diverse economic interests sharing a Network (Internet)

- browsers
- routers
- servers

Selfishness:

Parties deviate from their protocol if it is in their interest



Model Resulting Issues as

Games on Networks

Main question:
Quality of Selfish outcome

Well known: Central design can lead to better outcome than selfishness.

e.g.: Prisoner Dilemma

	C	D
C	2 2	1 99
D	99 1	98 98

Question: how much better?

Our Games

– Routing and Network formation: Users select paths that connects their terminals to minimize their own delay or cost

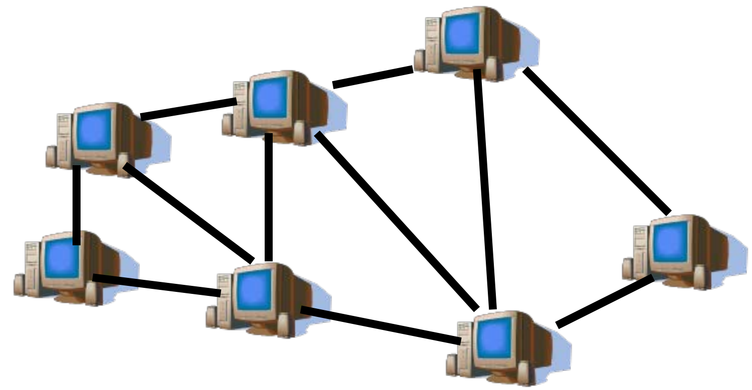
Why care about Games?

Users with a multitude of diverse economic interests sharing a Network (Internet)

- browsers
- routers
- servers

Selfishness:

Parties deviate from their protocol if it is in their interest



Model Resulting Issues as

Games on Networks

Some Games

- Routing:
- routers choose path for packets though the Internet
- Bandwidth Sharing:
- routers share limited bandwidth between processes
- Facility Location:
- Decide where to host certain Web applications
- Load Balancing
- Balancing load on servers (e.g. Web servers)
- Network Design:
- Independent service providers building the Internet

Main question:
Quality of Selfish outcome

Well known: Central design can lead to better outcome than selfishness.

e.g.: Prisoner Dilemma

	C	D
C	2 2	1 99
D	99 1	98 98

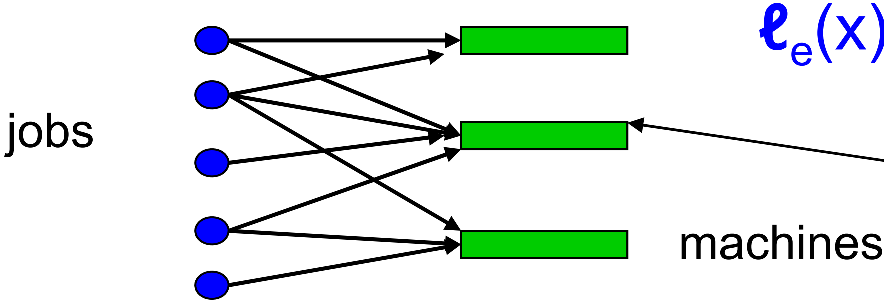
Question: how much better?

Our Games

– Routing and Network formation: Users select paths that connects their terminals to minimize their own delay or cost

load balancing and routing

Load balancing:



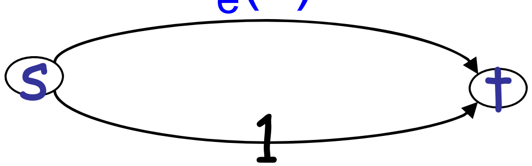
$\ell_e(x) = x$

Delay as a function of load:

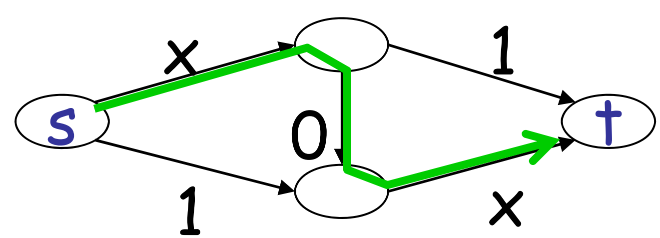
x unit of load $\rightarrow$ causes delay $\ell_e(x)$

Routing network:

$\ell_e(x) = x$



Allow more complex networks



Games: setup

- A set of players (in example: jobs)
- for each player, a set of strategies
  (which machine to choose)

Game: each player picks a strategy

For each strategy profile (a strategy for each player) → a payoff to each player
(load on selected machine)

Nash Equilibrium: stable strategy profile: where no player can improve payoff by changing strategy

Games: setup

Deterministic (pure) or randomized (mixed) strategies?

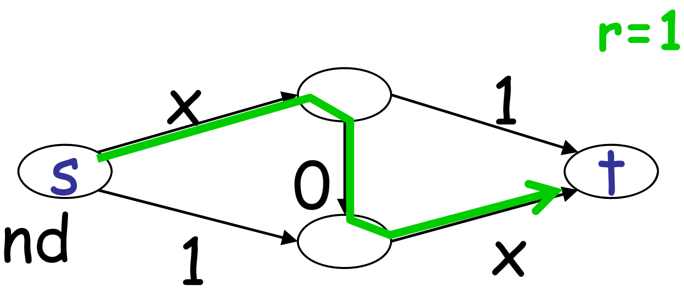
Pure: each player selects a strategy.

simple, natural, but stable solution may not exists

Mixed: each player chooses a probability distribution of strategies.

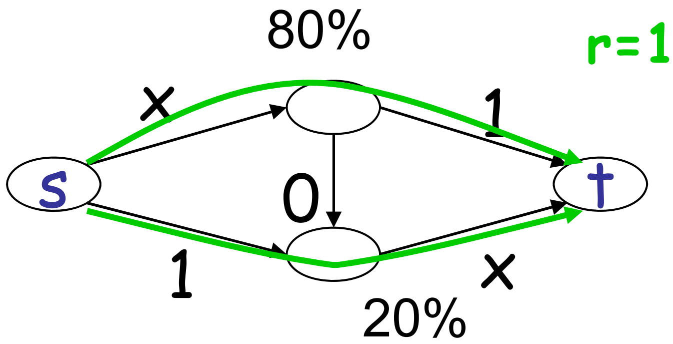
- equilibrium exists (Nash),
- but pure strategies often make more sense

Atomic vs. Non-atomic Game



Atomic Game:

- Each user controls a unit of flow, and
- selects a single path or machine



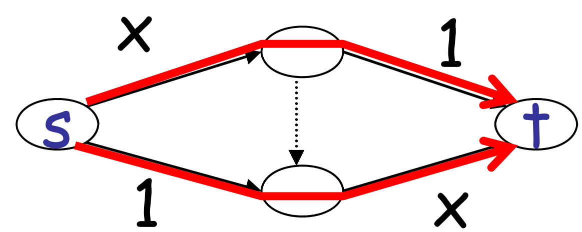
Non-atomic game:

- Users control an infinitesimally small amount of flow
- equilibrium: all flow path carrying flow are minimum total delay

Both congestion games: cost on edge e depends on the congestion (number of users)

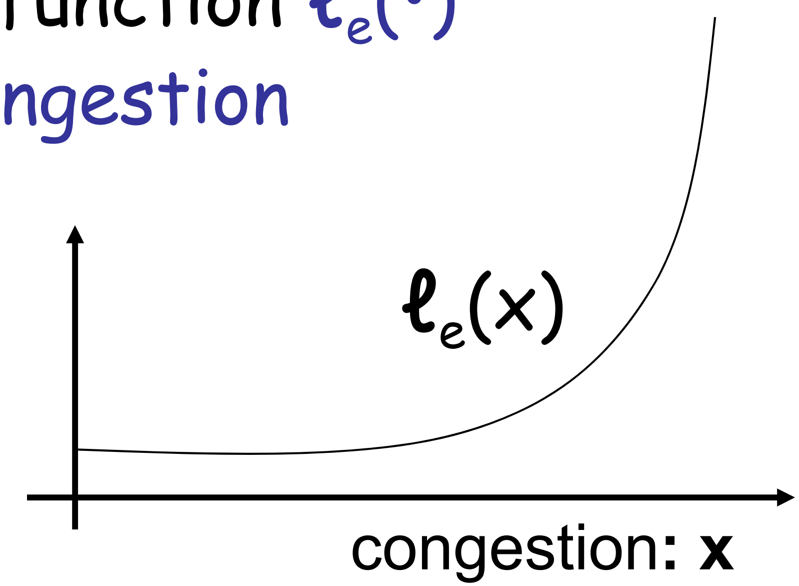
Congestion Games: Routing

- A directed graph $G = (V,E)$
- source-sink pairs s_i,t_i for $i=1,..,k$
- User i selects path P_i for traffic between s_i and t_i for each $i=1,..,k$

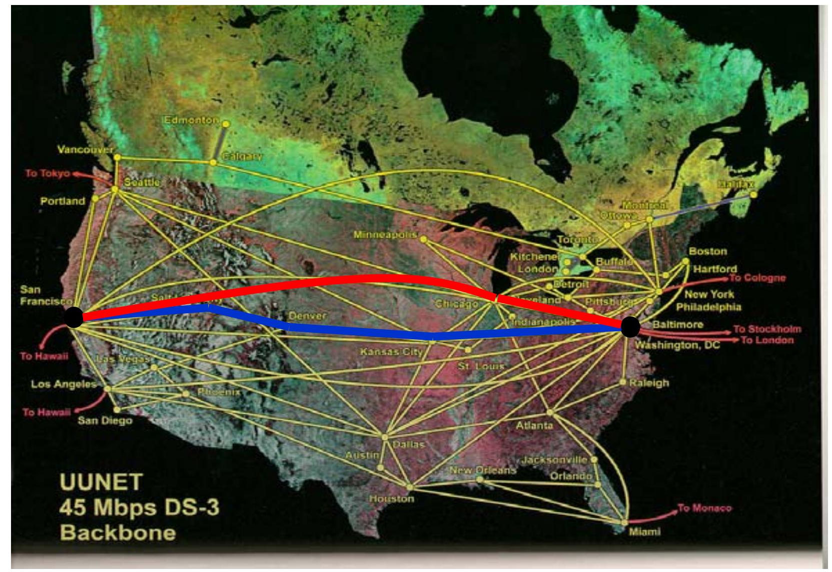


For each edge e a latency function $\ell_e(\cdot)$

Latency increasing with congestion



Example: Routing Game

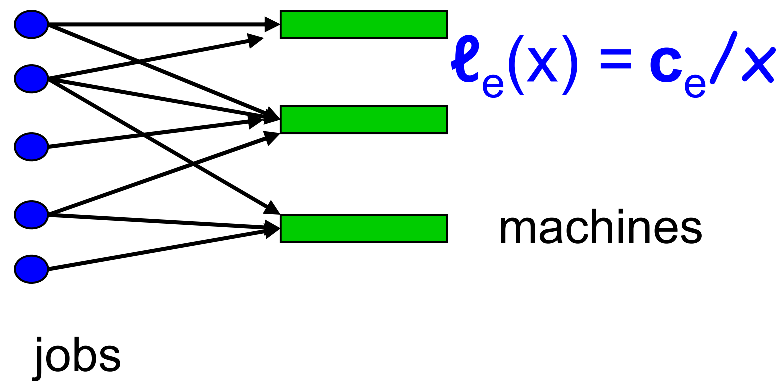


- Traffic subject to congestion delays
- cars and packets follow shortest path

Large number of participants!!

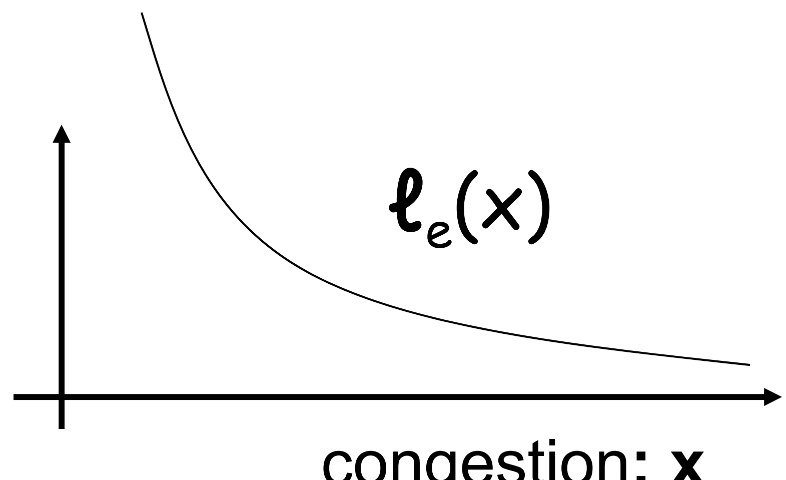
Congestion Games: Cost-sharing

- jobs $i=1,..,k$
- For each machine e a cost function $\ell_e(\cdot)$
  - E.g. cloud computing



- Cost decreasing with congestion (decreasing marginal cost)

$\ell_e(x) = \mathbf{c}_e/x$



Quality of Outcome:

Personal objective for player i:

load L_i or expected load $E(L_i)$

Overall objective?

- **Social Welfare:** $\sum_i L_i$ or expected value $E(\sum_i L_i)$

today

- **Alternately:** makespan: $\max_i L_i$ or 95% or the L_i values, etc
- With randomness: $\max_i E(L_i)$ or expected makespan $E(\max_i L_i)$

Connecting Nash and Opt

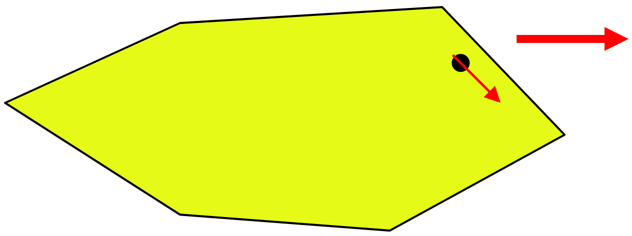
- Min-latency flow
  - for one s-t pair for simplicity

- minimize $C(f) = \Sigma_e f_e \cdot \ell_e(f_e)$
- subject to: f is an s-t flow
- carrying r units

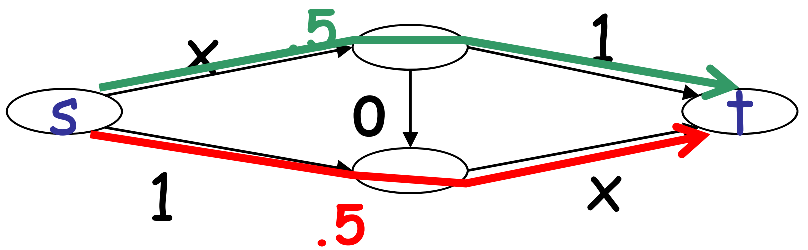
- By summing over edges rather than paths where f_e = amount of flow on edge e

Characterizing the Optimal Flow

Optimality condition: small change doesn't improve cost



For flows: all flow travels along minimum-gradient paths



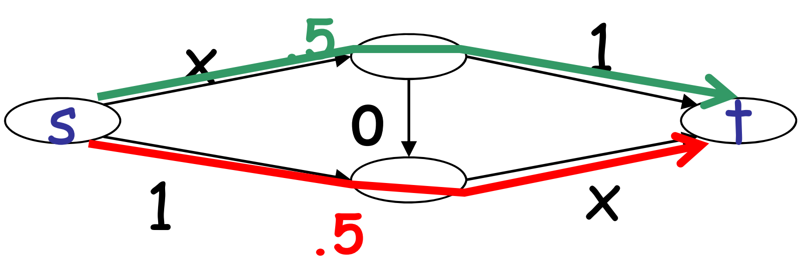
gradient is: $(x\,\ell(x))' = \ell(x) + x\,\ell'(x)$

selfish part   altruistic term

Optimal versus Nas Flow

Optimality condition: flow f is at minimum cost iff all flow travels along minimum-gradient paths: gradient is $\ell(x)+x\,\ell'(x)$

selfish part   altruistic term



Nash: flow f is at Nash equilibrium iff all flow travels along minimum-latency paths: $\ell(x)$

selfish part only

Nash $\leftrightarrow$ Min-Cost

Corollary 1: min cost is "Nash" with "delay"

$\ell(x)+x\,\ell'(x)$

Use of Corollary: If $x\,\ell'(x)$ is changed as tax, selfish users follow optimal paths

Nash $\leftrightarrow$ Min-Cost

Corollary 2: Nash is "min cost" with "cost"

$$\Phi(f) = \Sigma_e \int_0^{f_e} \ell_e(x)\, dx$$

Why?

gradient of $\Phi(f)$ is delay:

$$\left(\int_0^{f_e} \ell_e(x)\, dx\right)' = \ell(x)$$

Using function Φ

- Nash is the solution minimizing Φ

Theorem (Beckmann'56)

- In a network latency functions $\ell_e(x)$ that are monotone increasing and continuous,

- a deterministic Nash equilibrium exists, and is essentially unique

Using potential Φ…

- Nash minimizes the function Φ
- Hence,

$$\Phi(\text{Nash}) \leq \Phi(\text{OPT}).$$

Suppose that we also know for any solution

$$\Phi \leq \text{cost} \leq A\ \Phi$$

$\rightarrow$ cost(Nash) $\leq A\ \Phi$(Nash) $\leq A\ \Phi$(OPT) $\leq A$ cost(OPT).

$\rightarrow$ the Nash solution has good quality

Example: $\Phi \leq \text{cost} \leq A\,\Phi$

Example: $\ell_e(x) = x^d$ then

- total delay is $x \cdot \ell_e(x) = x^{d+1}$
- potential is $\int \ell_e(\xi)\, d\xi = x^{d+1}/(d+1)$

More generally: delay $\ell_e(x)$ degree d polynomial:

- ratio at most $d+1$

Sharp bound (see soon): price of anarchy for degree d polynomials is $O(d/\log d)$.

Sharper results for non-atomic games

Theorem 1 (Roughgarden-Tardos)

- In a network with linear latency functions
  - i.e., of the form $\ell_e(x)=a_e x+b_e$
- the cost of a Nash flow is at most 4/3 times that of the minimum-latency flow

However, O(d/log d) large as degree d gets large…

Aside: for non-atomic games

Theorem 3 (Roughgarden-Tardos):

- In any network with continuous, nondecreasing latency functions

cost of Nash with rates r_i for all i	$\leq$	cost of opt with rates $2r_i$ for all i

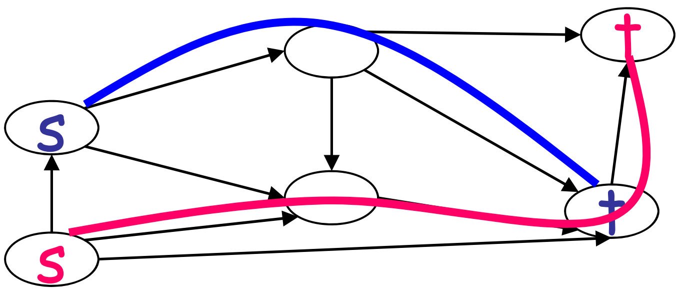
Proof idea:

Opt may cost very little, but marginal cost is as high as latency in Nash

$\rightarrow$ Augmenting to double rate costs at least as much as Nash

Atomic (discrete) Analog

- Each user controls one unit of flow, and
- selects a single path



Theorem Change in potential is same as function change perceived by one user [Rosenthal'73, Monderer Shapley'96,]

$$\Phi(f) = \Sigma_e (\ell_e(1)+\ldots+ \ell_e(f_e)) = \Sigma_e \Phi_e$$

Even though moving player ignores all other users

[Recall continuous potential: $\Phi(f) = \Sigma_e \int_0^{f_e} \ell_e(x)\, dx$]

Corollary: Nash equilibria are local min. of $\Phi(f)$

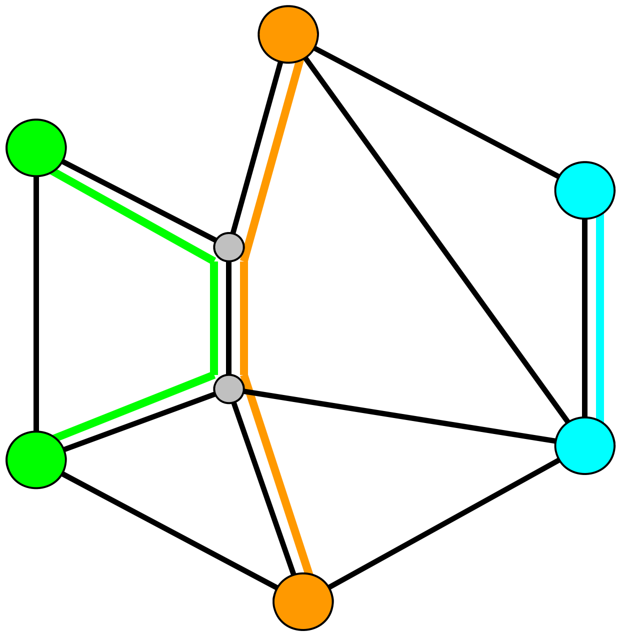
Network Design as Potential Game

Given: $G = (V,E)$,
costs $c_e(x)$ for all $e \in E$,
k terminal sets (colors)

Have a player for each color.

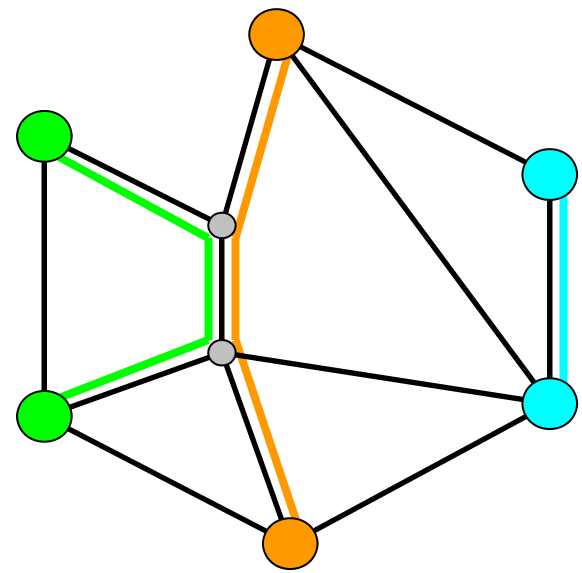
Each player wants to build a network in which his nodes are connected.

Player strategy: select a tree connecting his set.



Costs in Connection Game

Players pay for their trees, want to minimize payments.

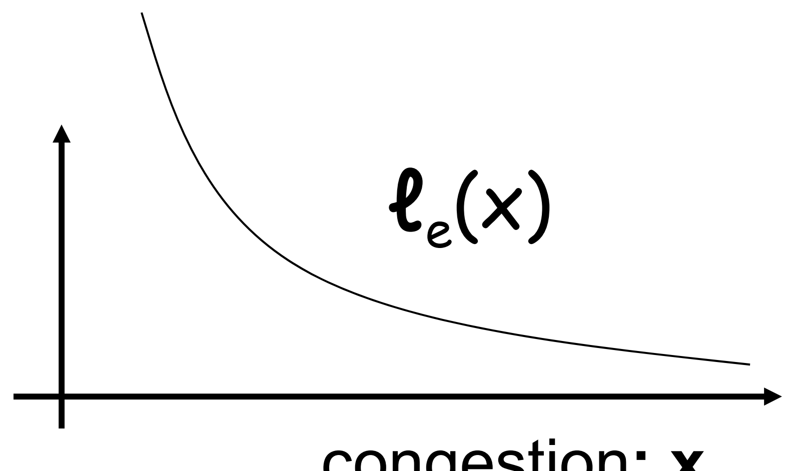


What is the cost of the edges?

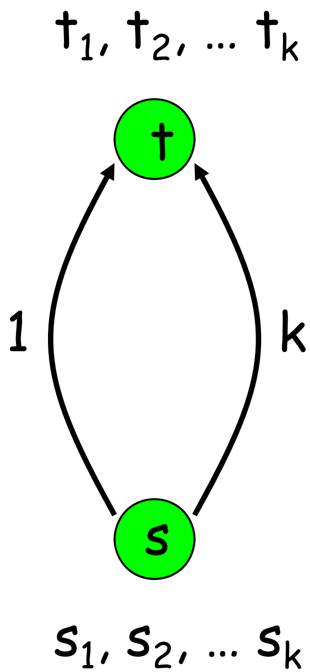
$c_e(x)$ is cost of edge e for x users.

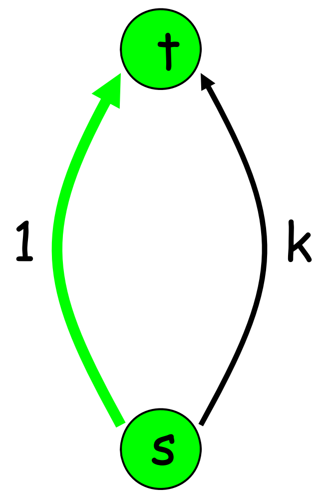
Assume economy of scale and fair sharing:

e.g.: $\ell_e(x) = c_e(x) / x$

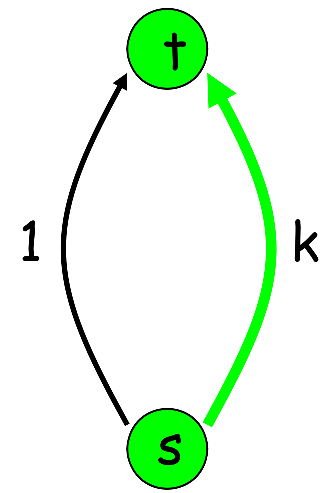


A Simple Example





One NE:
each player
pays 1/k



Another NE:
each player
pays 1

Results for Network Design

Theorem [Anshelevich, Dasgupta, Kleinberg, Tardos, Wexler, Roughgarden FOCS'04]

There exists equilibrium with cost $\leq$ O(log k)Opt for k players (bound sharp)

Proof

$$\text{cost} \leq \Phi \leq \text{cost} \cdot O(\log k)$$

$$\text{Price of Stability} = \frac{\text{cost of best selfish outcome}}{\text{“socially optimum” cost}}$$

Design with constraint for stability

Stronger proof technique

- bounds price of anarchy (not stability)
- Tight bounds in many games

A game is (λ,μ)-smooth if, for every pair f, f^* outcomes ($\lambda > 0$; $\mu < 1$):

$$\Sigma_e f^*_e \cdot \ell_e(f_e) \leq \lambda \Sigma_e f^*_e \cdot \ell_e(f^*_e) + \mu \Sigma_e f_e \cdot \ell_e(f_e)$$

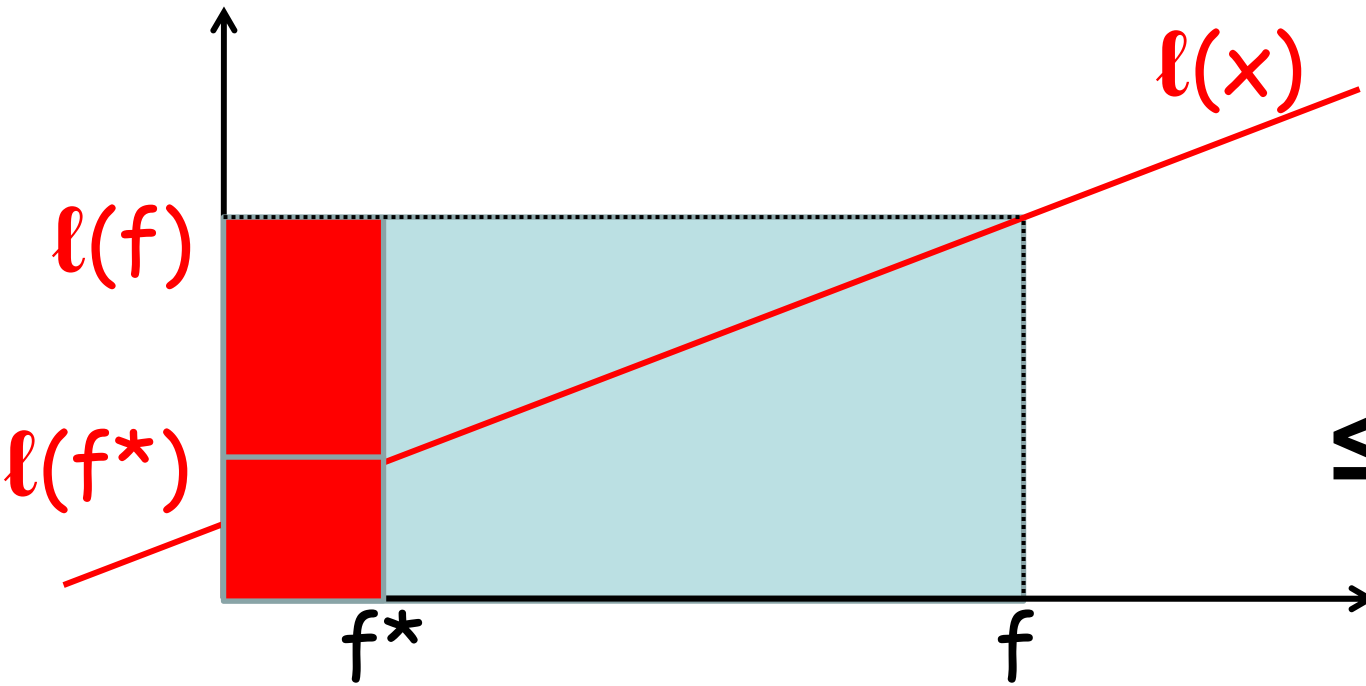
or for all $f, f^* \geq 0$

$$f^* \cdot \ell(f) \leq \lambda f^* \cdot \ell(f^*) + \mu f \cdot \ell(f)$$

Linear delay is smooth

Claim: $f^* \cdot \ell(f) \leq f^* \cdot \ell(f^*) + \frac{1}{4} f \cdot \ell(f)$

assuming $\ell(f)$ linear: $\lambda = 1$; $\mu = \frac{1}{4}$



$f^* \ell(f)$
$- f^* \ell(f^*)$
$\leq \frac{1}{4} f \, \ell(f)$

Discrete version

Smooth for flows:

$\Sigma_e f^*_e \cdot \ell_e(f_e) \leq \lambda \Sigma_e f^*_e \cdot \ell_e(f^*_e) + \mu \Sigma_e f_e \cdot \ell_e(f_e)$

A game is (λ,μ)-smooth if, for every pair s,s^* outcomes

$\Sigma_i C_i(s^*_i, s_{-i}) \leq \lambda \text{ cost}(s^*) + \mu \text{ cost}(s)$

Where $\text{cost}(s) = \Sigma_i C_i(s)$

s_i strategy of user i

s_{-i} strategies of all users

Smooth ⇒ Price of Anarchy [Roughgarden]

Use smooth for s = Nash and s* = opt

$\Sigma_i\, C_i(s^*_i, \mathbf{s}_{-i}) \le \boldsymbol{\lambda}\ \mathbf{cost(s^*)} + \boldsymbol{\mu}\ \mathbf{cost(s)}$

$\text{cost}(s) = \Sigma_i\, C_i(s_i, \mathbf{s}_{-i})$

$\le \Sigma_i\, C_i(s^*_i, \mathbf{s}_{-i})$ [s a Nash eq]

$\le \lambda\ \text{cost}(s^*) + \mu\ \text{cost}(s)$ [smooth]

Then: $\text{cost}(s) \le \lambda/(1-\mu)\ \text{cost}(s^*)$

Atomic Smoothness Bound

atomic linear delay smooth

$\sum_i C_i(f^*_i, f_{-i}) \leq \lambda\ \mathbf{cost}(f^*) + \mu\ \mathbf{cost}(f)$

Consider edge by edge:

(nonatomic version):

$f^* \cdot \ell(f) \leq \lambda\ f^* \cdot \ell(f^*) + \mu\ f \cdot \ell(f)$

Atomic version

$f^* \cdot \ell(f+1) \leq \lambda\ f^* \cdot \ell(f^*) + \mu\ f \cdot \ell(f)$

basic inequality: $y(z+1) \leq (5/3)y^2 + (1/3)z^2$

Implicit Smoothness Bounds

Examples: selfish routing, linear cost fns.

- every nonatomic game is (1,1/4)-smooth
  - follows directly from analysis in [Correa/Schulz/Stier Moses 05]
- every atomic game is (5/3,1/3)-smooth
  - follows directly from analysis in [Awerbuch/Azar/Epstein 05], [Christodoulou/Koutsoupias 05]
  - Implies a 5/2 bound on Price of Anarchy

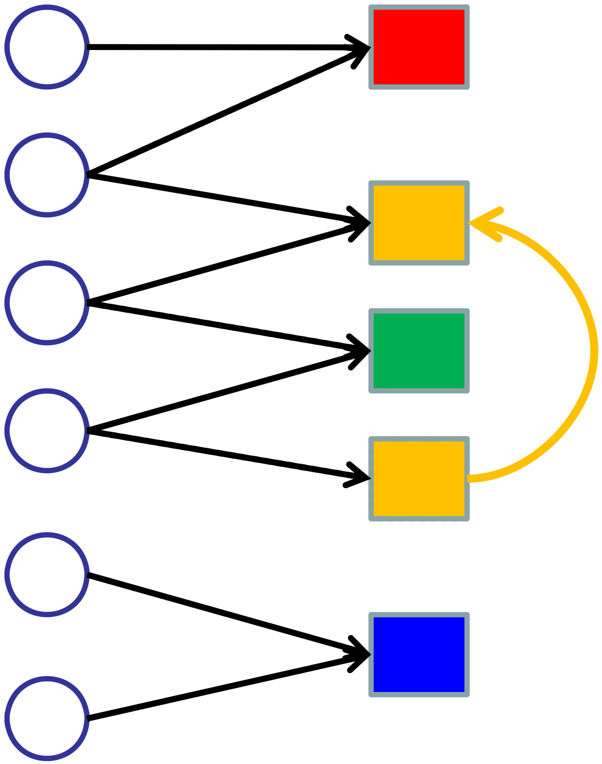
Theorem [Roughgarden 09] for congestion game the best such bound tight

Smoothness for Value Problems

Vetta "competitive societies": value for facility location:

s Nash, s* Optimum

$\mathrm{Val}(s) \geq \Sigma_i \, \mathrm{Val}_i\,(s^*_i, s_{-i}) \geq \mathrm{Val}(s^*) - \mathrm{Val}(s)$



hence $\mathrm{Val}(S) \geq \frac{1}{2}\mathrm{Val}(s^*)$.

fyi: Also a potential game

Summary

Congestion games are potential games

- $\exists$ Pure equilibria (min of potential)
- Min of potential has OK quality
- Price of stability (or anarchy when unique)
- Smoothness and stronger Price of anarchy bounds
  - Applies to some other games also

Tomorrow:

- Learning in games (why and how?)
- solutions reached via learning