Selfish Routing and the Price of Anarchy

Tim Roughgarden
Cornell University

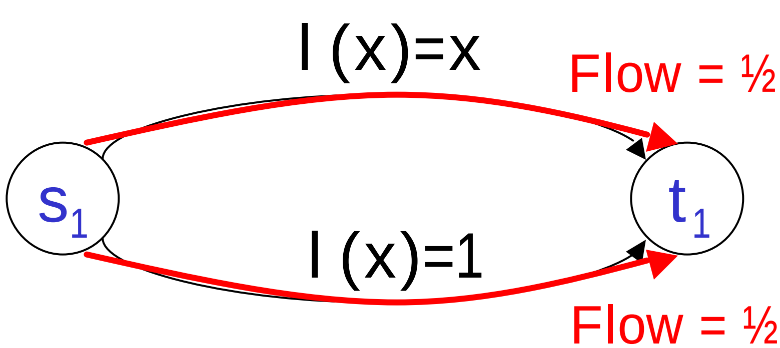
Includes joint work with Éva Tardos

Traffic in Congested Networks

The Model:

- A directed graph $G = (V,E)$
- k source-destination pairs $(s_1, t_1), \ldots, (s_k, t_k)$
- A rate r_i of traffic from s_i to t_i
- For each edge e, a latency function $l_e(\bullet)$ [cts, nondecreasing]

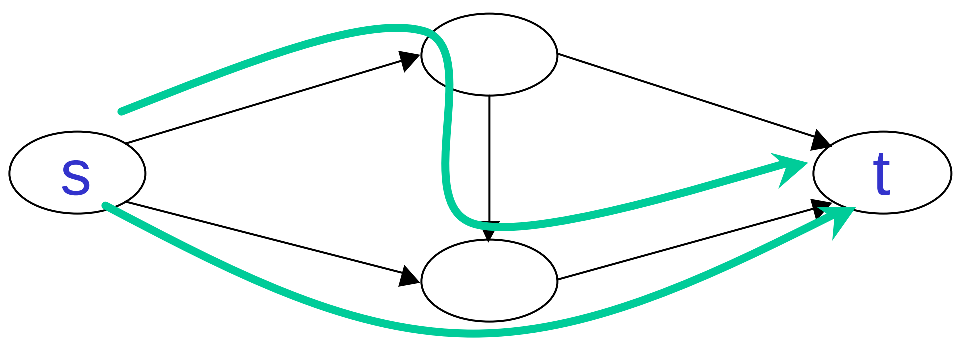
Example: $(k, r=1)$



Selfish Routing

Traffic and Flows:

- f_P = amount of traffic routed on s_i-t_i path P
- flow vector f ⇔ routing of traffic



Selfish routing: what flows arise as the routes chosen by many noncooperative agents?

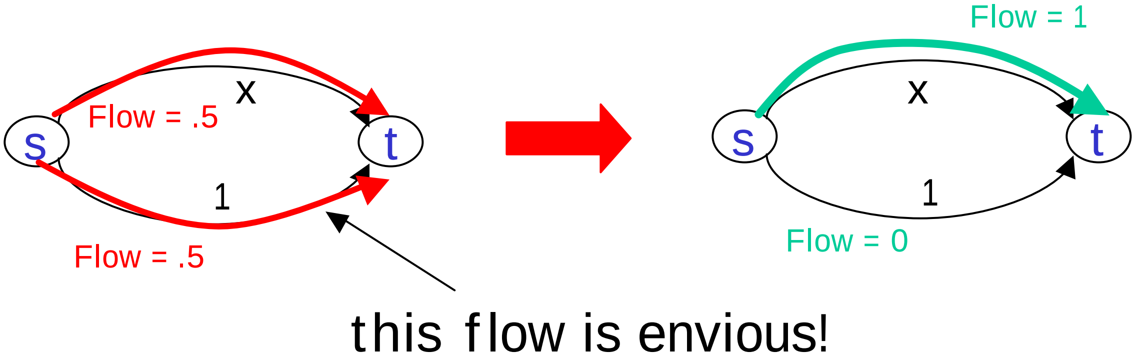
Nash Flows

Some assumptions:

- agents small relative to network
- want to minimize personal latency

Def: A flow is at Nash equilibrium (or is a Nash flow) if all flow is routed on min-latency paths [given current edge congestion]

Example:



Some History

- traffic model, def of Nash flows due to [Wardrop 52]
  - historically called user-optimal/user equilibrium

- Nash flows always exist, are (essentially) unique
  - due to [Beckmann et al. 56]

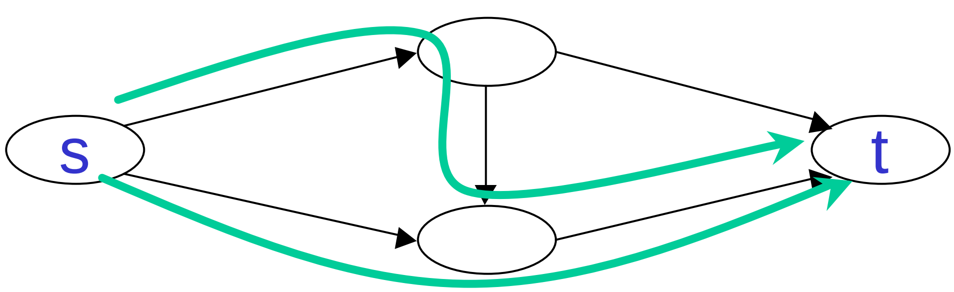
The Cost of a Flow

Our objective function:

- $l_P(f)$ = sum of latencies of edges of P (w.r.t. the flow f)

- $C(f)$ = cost or total latency of flow f: $S_P \, f_P \bullet l_P(f)$

  also: $S_e \, f_e \bullet l_e(f_e)$

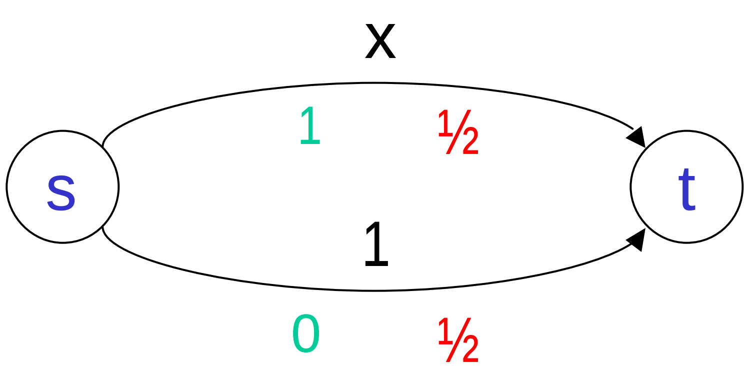


Central question: how good (or bad) are Nash flows?

The Inefficiency of Nash Flows

Fact: Nash flows do not optimize total latency [Pigou 1920]

- Þ lack of coordination leads to inefficiency



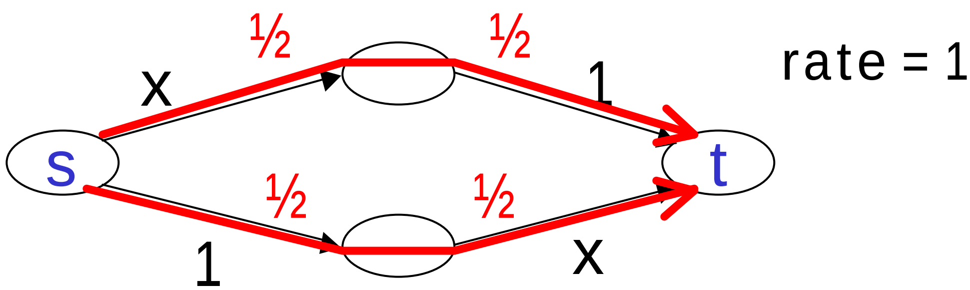
Cost of Nash flow = 1•1 + 0•1 = 1

Cost of optimal (min-cost) flow

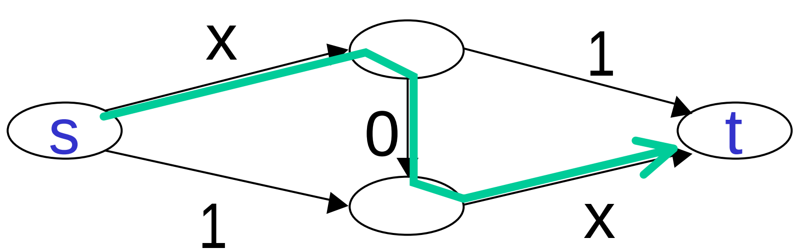
= ½•½ +½•1 = ¾

Braess's Paradox

Better network, worse Nash flow:



Cost of Nash flow = 1.5



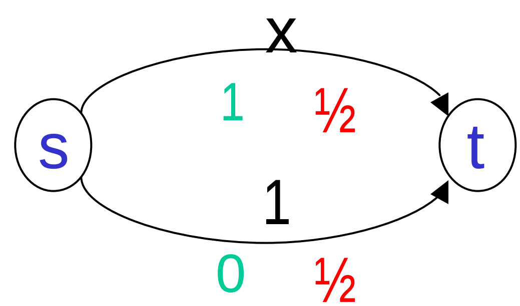
Cost of Nash flow = 2

All traffic incurs more latency!

- due to [Braess 68]
- see also [Roughgarden 01]

How Bad is Selfish Routing?

Pigou's example is simple...



How inefficient are Nash flows:

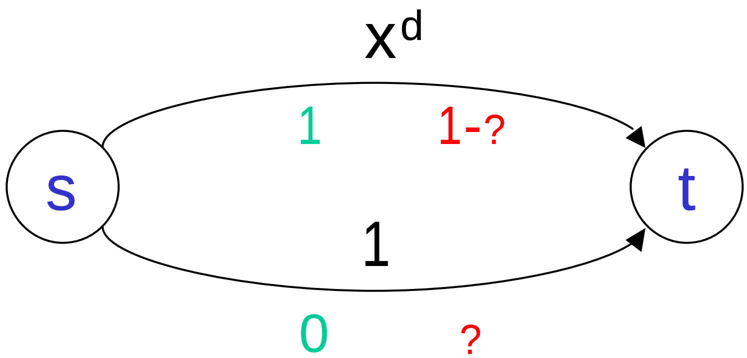
- with more realistic latency fns?
- in more realistic networks?

Goal: prove that Nash flows are near-optimal

- want a laissez-faire approach to managing networks
  - also [Koutsoupias/Papadimitriou 99]

The Bad News

Bad Example: (r = 1, d large)



Nash flow has cost 1, min cost ≈ 0

Þ Nash flow can cost arbitrarily more than the optimal (min-cost) flow

- even if latency functions are polynomials

A Bicriteria Bound

Approach #1: settle for weaker type of guarantee

Theorem: [Roughgarden/Tardos 00] network w/cts, nondecreasing latency functions Þ

cost of Nash at rate r	=	cost of opt at rate 2r

Corollary: M/M/1 delay fns (l(x)=1/(u-x), u = capacity) Þ

Nash cost w/ capacities 2u	=	opt cost w/ capacities u

Linear Latency Functions

Approach #2: restrict class of allowable latency functions

Def: a linear latency function is of the form $l_e(x)=a_e x+b_e$

Theorem: [Roughgarden/Tardos 00]

network w/linear latency fns Þ

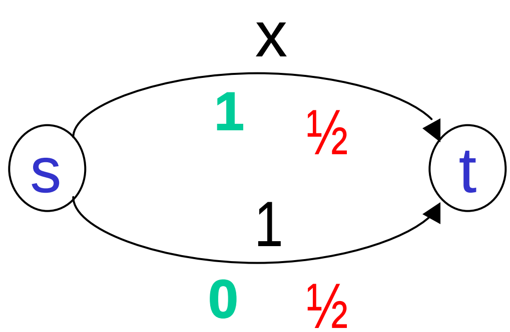
cost of Nash flow = 4/3 × cost of opt flow

aka price of anarchy
[Papadimitriou 01]

Sources of Inefficiency

Corollary of main Theorem:

- For linear latency fns, worst Nash/OPT ratio is realized in a two-link network!



- Cost of Nash = 1
- Cost of OPT = ¾

- one source of inefficiency:
  - confronted w/ two routes, selfish users overcongest one of them
- Corollary ⇒ that's all, folks!
  - network topology plays no role

No Dependence on Network Topology

Thm: [Roughgarden 02] for any class of convex latency fns including the constant fns, worst Nash/OPT ratio occurs in a two-node, two-link network.

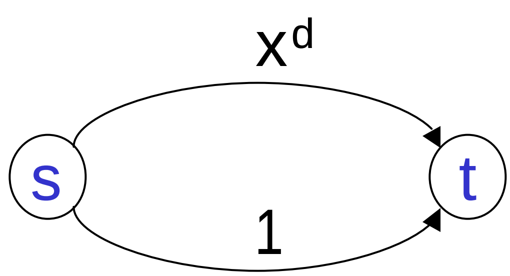
- inefficiency of Nash flows always has simple explanation
- network topology plays no role

Recall: worst ratio may be (much) larger than 4/3 (modify Pigou's ex)

Computing the Price of Anarchy

Application: worst-case examples simple Þ worst-case ratio is easy to calculate

Example: polynomials with degree = d, nonnegative coeffs Þ price of anarchy T(d/log d)



Also: M/M/1, M/G/1 queue delay fns, etc.

Comparison to Previous Work

Remark: parallel links are not worst-case examples for:

- Approximate Nash flows, integral Nash flows [Roughgarden/Tardos FOCS '00]
- Stackelberg equilibria [Roughgarden STOC '01]
- Braess's paradox, maximum travel time obj fn [Roughgarden FOCS '01]
- Equilibria w/explicit capacities [Schulz/Stier SODA '03]

The Bicriteria Bound

Theorem: [Roughgarden/Tardos 00]
network w/cts, nondecreasing latency functions Þ

cost of Nash at rate r	=	cost of opt at rate 2r

Corollary: M/M/1 delay fns (l(x)=1/(u-x), u = capacity) Þ

Nash cost w/ capacities 2u	=	opt cost w/ capacities u

Key Difficulty

Sps f a Nash flow, f^* an opt flow at twice the rate.

Recall: we can write

$$C(f) = \Sigma_e f_e \cdot l_e(f_e)$$

- sum over edges instead of paths
- f_e = amount of flow on edge e

Similarly: $C(f^*) = \Sigma_e f_e^* \cdot l_e(f_e^*)$

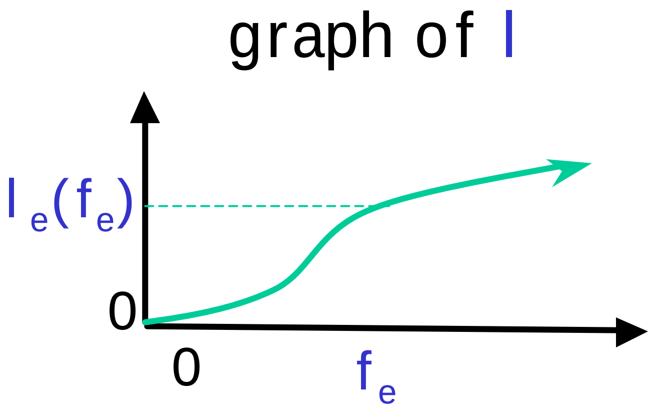
Problem: what is the relation between $l_e(f_e)$ and $l_e(f_e^*)$?

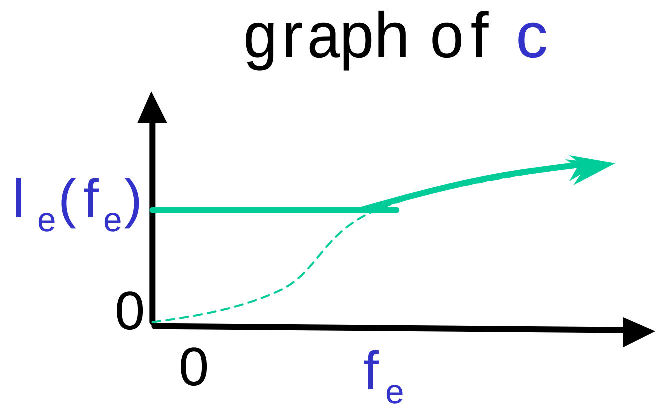
Key Trick

Idea: lower bound cost of f^* using a different set of latency fns c with the properties:

- easy to lower bound cost of f^* w.r.t. latency fns c
- cost of f^* w.r.t. latency fns c $\approx$ cost of f^* w.r.t. latency fns l

The construction:

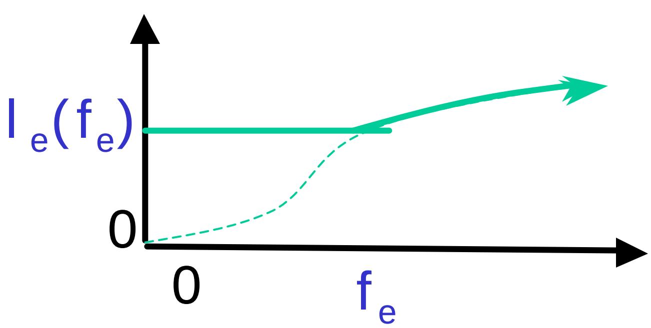




Lower Bounding OPT

Assume: only one commodity (multicommodity no harder).

Key observation: latency of path P w.r.t. latency fns c with no congestion is $l_P(f)$ [latency in Nash]



Corollary: Suppose in Nash, everyone has latency L. Then:

- cost of f^* w.r.t. c is $\geq 2rL$
- $C(f) = rL$.

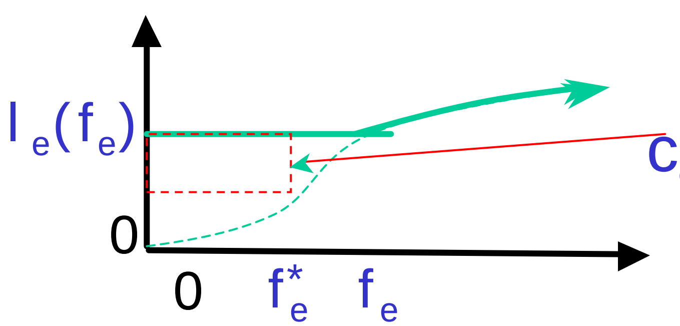
Upper Bounding the Overestimate

Thus: cost of f^* w.r.t. c is $\geq 2C(f)$.

Claim: (will finish proof of Thm)

[cost of f^* w.r.t. c] $- C(f^*) = C(f)$.

Reason: difference in costs on e is



typical value of $c_e(f_e^*)f_e^* - l_e(f_e^*)f_e^*$

Þ $c_e(f_e^*)f_e^* - l_e(f_e^*)f_e^* = l_e(f_e)f_e$

sum over edges to get Claim

Summary

Goal: prove that loss in network performance due to selfish routing is not too large.

Problem: a Nash flow can cost arbitrarily more than an optimal flow.

Solutions:

- prove a bicriteria bound instead
- restrict class of allowable edge latency functions

Nonatomic Congestion Games

Question: in what other games is the outcome of selfishness near-optimal?

Thm: [Roughgarden/Tardos 02] All results from this talk generalize to nonatomic congestion games:

- replace network by ground set
- s_i-t_i paths by set systems