New Perspectives and Challenges in Routing Games: Query models & Signaling

Chaitanya Swamy
University of Waterloo

New Perspectives and Challenges in Routing Games: Query models ~~& Signaling~~

Chaitanya Swamy
University of Waterloo

Mostly based on joint work with

Umang Bhaskar	TIFR
Katrina Ligett	Hebrew University & Caltech
Leonard Schulman	Caltech

Routing games

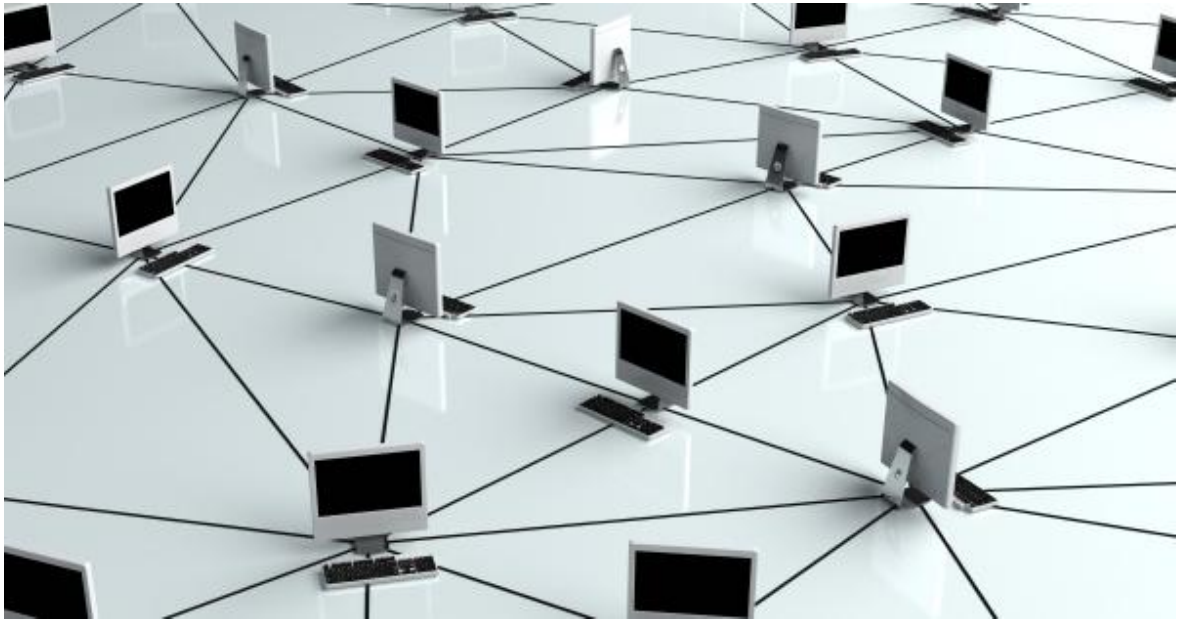
- Model for traffic in networks

Routing games

- Model for traffic in networks, e.g.,
  - road networks

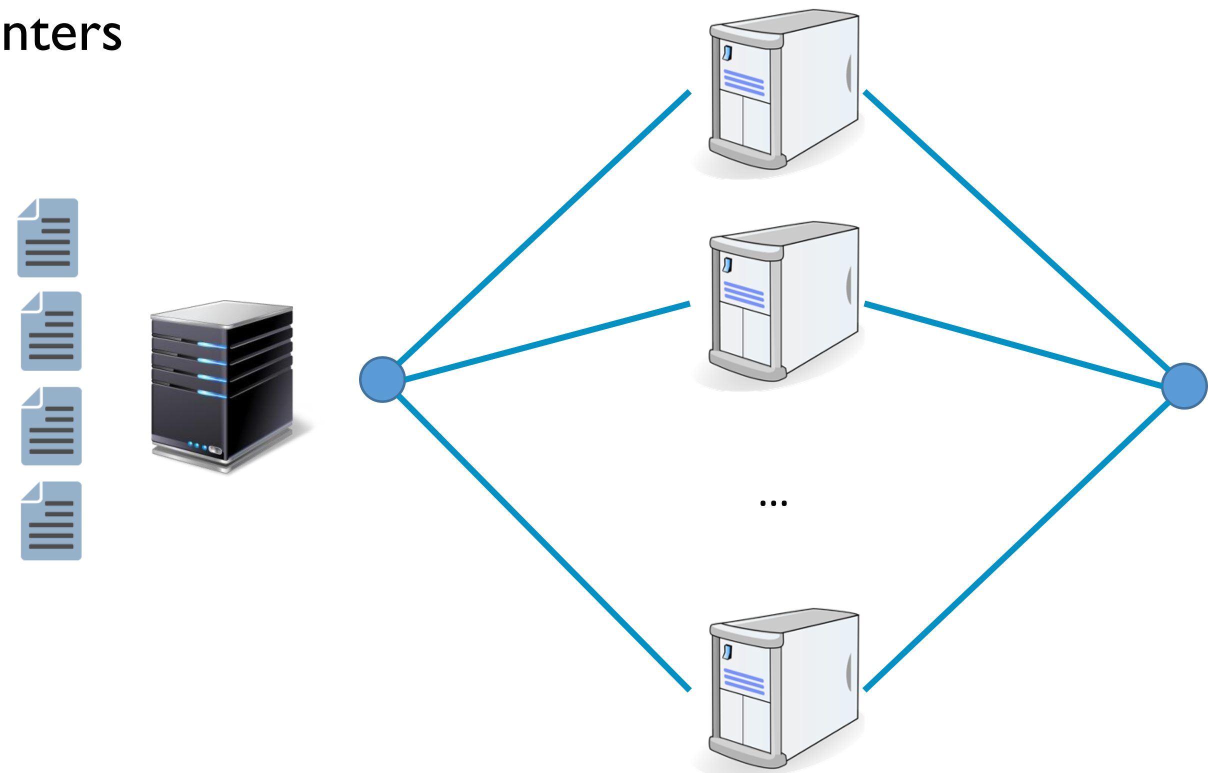
Routing games

- Model for traffic in networks, e.g.,
  - road networks
  - data networks



Routing games

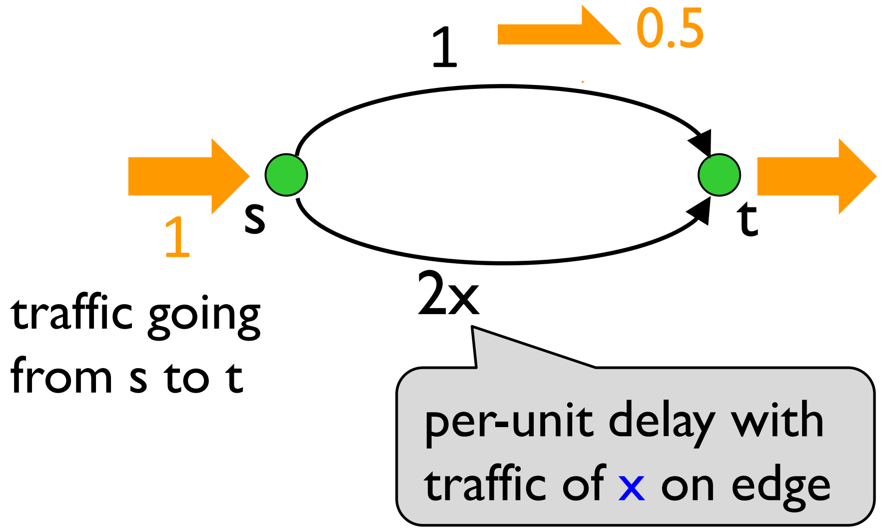
- Model for traffic in networks, e.g.,
  - road networks
  - data networks
  - jobs in data centers



Routing games

- Model for traffic in networks, e.g.,
  - road networks
  - data networks
  - jobs in data centers
- Common features:
  - resources (e.g., roads) shared across various agents (players)
  - nobody dictates use of resources
  - players compete for resources
- Routing games: game-theoretic model for traffic in networks
- Seek to reason about how competition affects traffic

Routing games: mathematical model



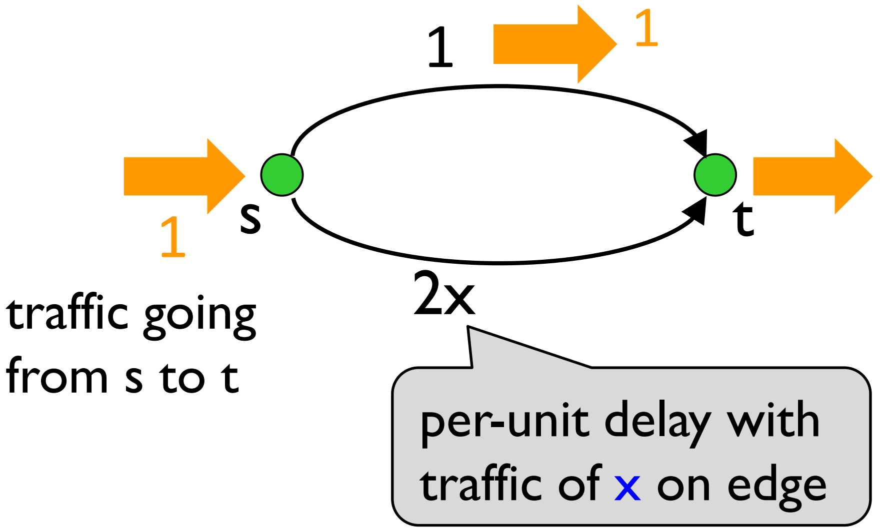
Players control infinitesimal traffic

Choose route from s to t

⇒ get an s-t flow of volume 1

Called nonatomic routing

Routing games: mathematical model



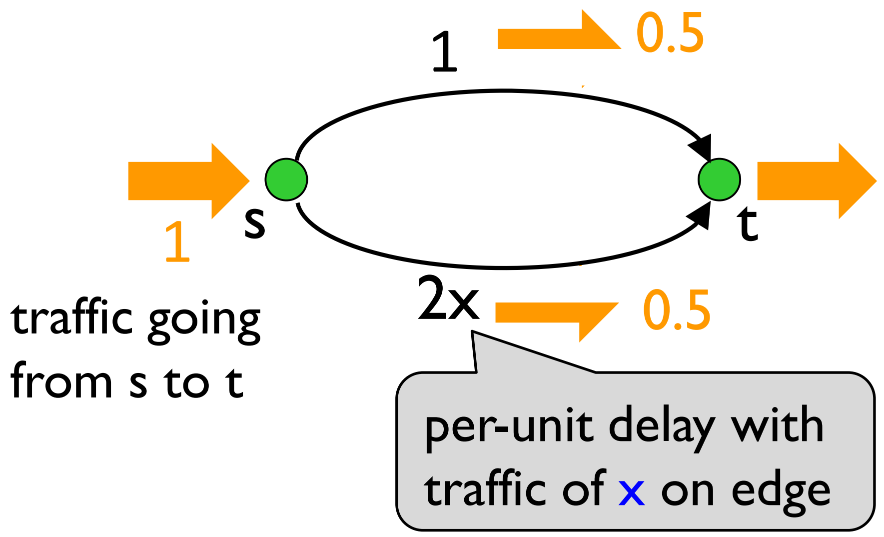
Players control infinitesimal traffic

Choose route from s to t

⇒ get an s-t flow of volume 1

Called nonatomic routing

Routing games: mathematical model



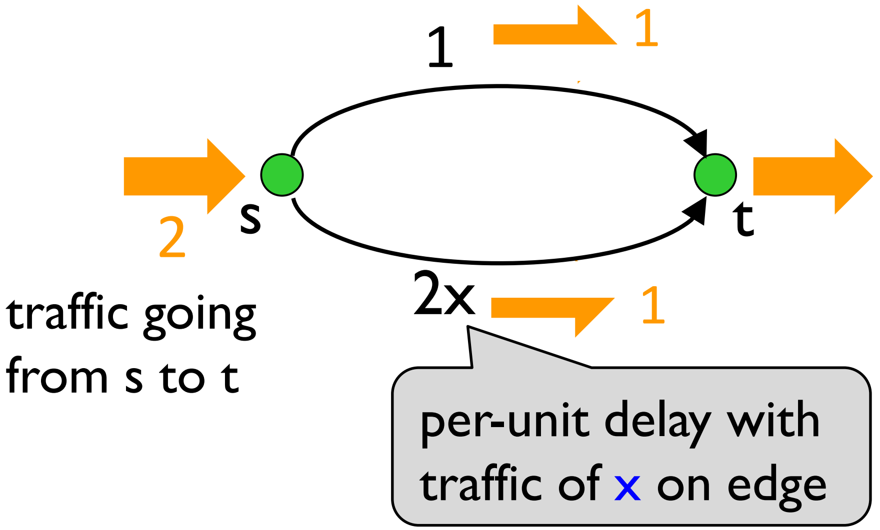
Players control infinitesimal traffic

Choose route from s to t

$\Rightarrow$ get an s-t flow of volume 1

Called nonatomic routing

Routing games: mathematical model



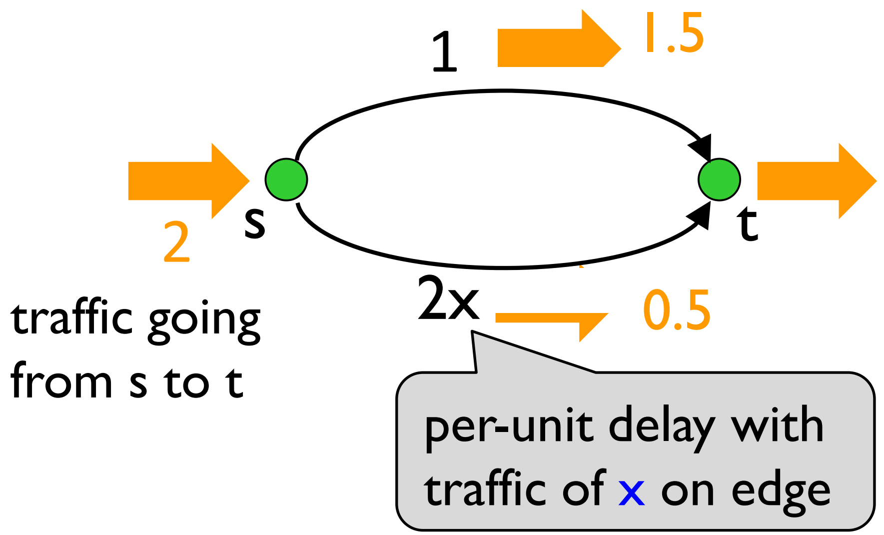
Players control infinitesimal traffic

Choose route from s to t

$\Rightarrow$ get an s-t flow of volume 2

Called nonatomic routing

Routing games: mathematical model



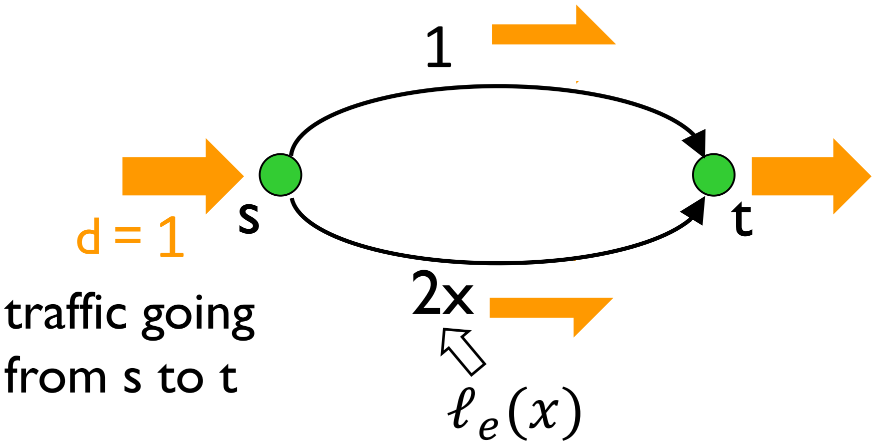
Players control infinitesimal traffic

Choose route from s to t

$\Rightarrow$ get an s-t flow of volume 2

Called nonatomic routing

Routing games: mathematical model



Players control infinitesimal traffic

Choose route from s to t

$\Rightarrow$ get an s-t flow of volume 1

Called nonatomic routing

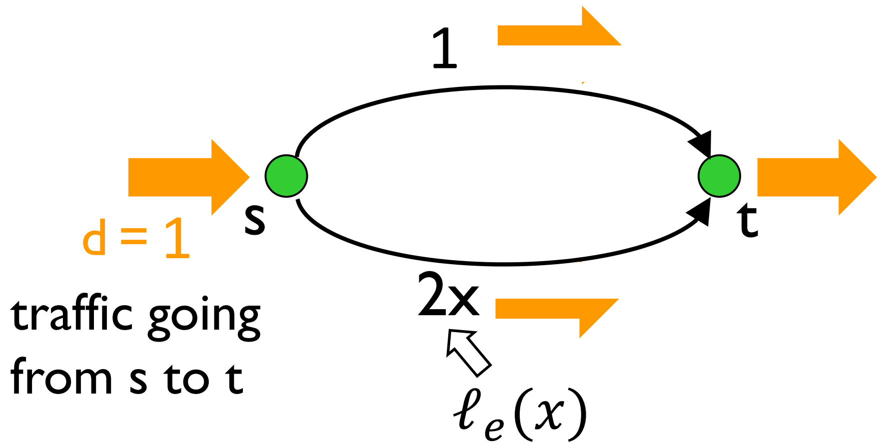
Equilibrium: each player chooses least-delay route given other players' choices

Formally, a nonatomic routing game is specified by

$$\Gamma = (G, \{\ell_e : \mathbb{R}_+ \mapsto \mathbb{R}_+\}_e, s, t, d)$$

- G: directed graph
- $\{\ell_e\}_e$: edge latency functions (will assume are ↑)
- s, t, d: source, sink, demand

Routing games: mathematical model



Players control infinitesimal traffic

Choose route from s to t

$\Rightarrow$ get an s-t flow of volume 1

Called nonatomic routing

Equilibrium: each player chooses least-delay route given other players' choices

Formally, a nonatomic routing game is specified by

$$\Gamma = (G, \{\ell_e: \mathbb{R}_+ \mapsto \mathbb{R}_+\}_e, s, t, d)$$

An s-t flow f of volume d is an equilibrium flow $\Leftrightarrow$

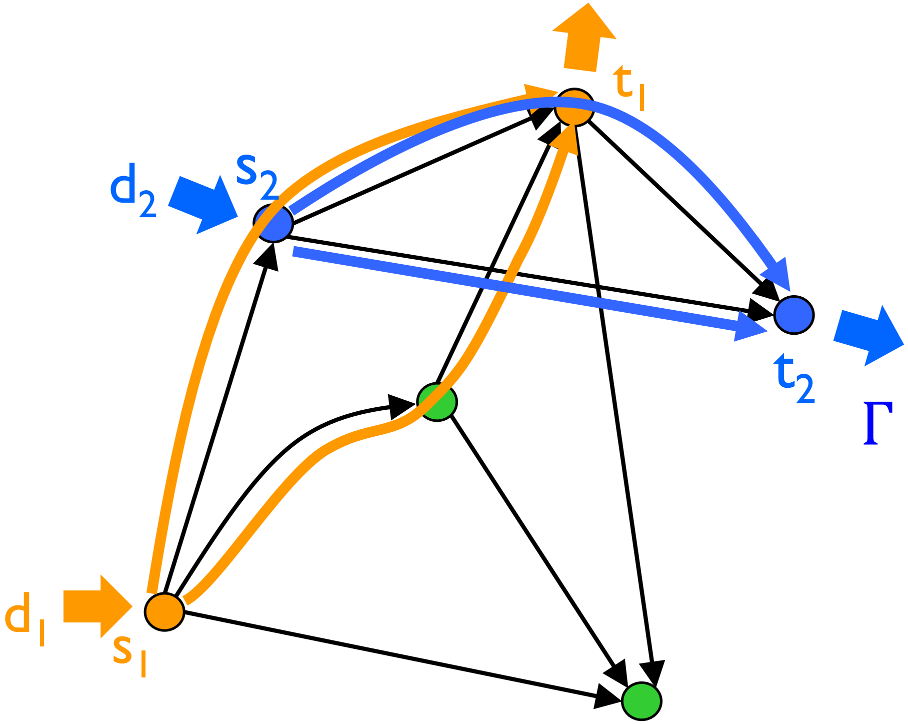
for all s-t paths P, Q with $f_e > 0\ \forall e \in P$,

$$\sum_{e \in P} \ell_e(f_e) \leq \sum_{e \in Q} \ell_e(f_e)$$

total delay along P

total delay along Q

Routing games: mathematical model



More generally, could have many (source, sink, demand) tuples called commodities:

$$\Gamma = (G, \{\ell_e\colon \mathbb{R}_+ \mapsto \mathbb{R}_+\}_e, \{s_i, t_i, d_i\}_{i=1}^k)$$

Model dates back to Wardrop 1952, Beckmann-McGuire-Winston 1956

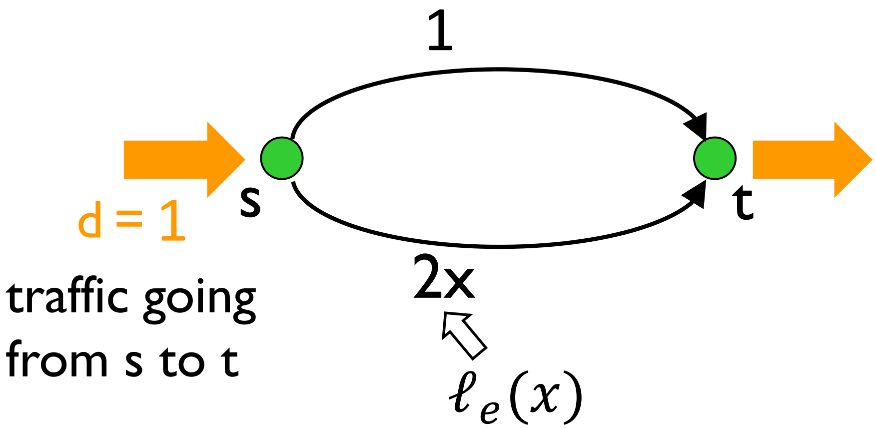
Equilibrium notion due to Wardrop

A multicommodity flow $f = (f_1, \ldots, f_k)$, where each f^i routes d_i flow from s_i to t_i is an equilibrium flow $\Leftrightarrow$

for all s_i-t_i paths P, Q with $f_e^i > 0\ \forall e \in P$,

$$\sum_{e \in P} \ell_e(f_e) \leq \sum_{e \in Q} \ell_e(f_e)$$

Routing games: mathematical model



finite amounts of

Players control ~~infinitesimal~~ traffic

Called ~~no~~natomic routing

Choose how to route their demand from their source to sink

- 1 route: atomic unsplittable
- multiple routes: atomic splittable

minimum-delay routing of its demand

Equilibrium: each player chooses ~~least delay route~~

given other players' choices

Some basic questions

- Does equilibrium flow exist? Is it unique?
- Can an equilibrium be computed efficiently?
  - In a decentralized way by players' moves?
- How bad are equilibria wrt. optimal flows?
  - inefficiency of worst equilibrium: price of anarchy
  - Inefficiency of best equilibrium: price of stability
- Equilibria may be undesirable:
  - large total delay compared to optimal flow
  - heavy traffic in undesirable regions (e.g., residential areas)

  Can one steer equilibria to desirable flows? (E.g., by imposing tolls on edges, or controlling portion of total flow)

For nonatomic routing

- Beckman et al. '56: Equilibria always exist, can be computed efficiently by solving:

  Minimize $\sum_e \int_0^{f_e} \ell_e(x)dx$

  s. t. $f = \sum_i f^i$, f^i routes d_i flow from s_i to t_i

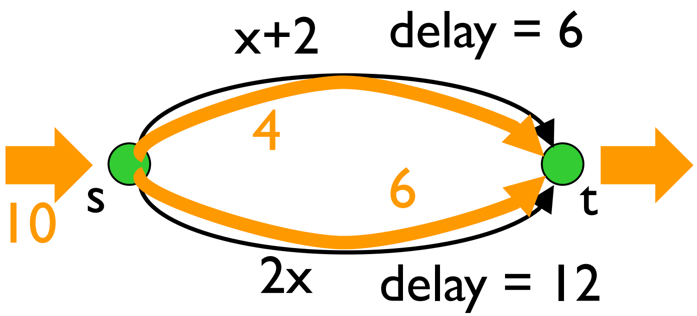
  All $\ell_e(x) \uparrow \Rightarrow$ strictly convex program $\Rightarrow$ unique equilibrium

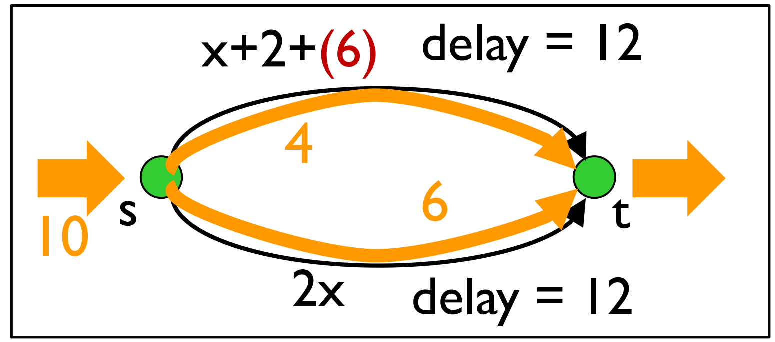
- Roughgarden-Tardos '02, Roughgarden '03: Total delay of equilibrium can be much worse than that of optimal flow. Can give a formula for (worst-case) price of anarchy for any class of latency functions (under mild conditions).

For nonatomic routing

- Can efficiently find tolls on edges (if they exist) so tha resulting equilibrium is a given target flow (e.g., optim

Equilibrium after tolls





toll τ_e on edge e changes "delay" on e to $\ell_e(x) + \tau_e$ ⇨ cost
(assuming here that players value time and money equally)

At equilibrium, players choose least-cost paths

Any minimal target flow f^* can be imposed via edge tolls.
The tolls can be computed by solving an LP.

(Beckmann et al.'56, Cole et al.'03, Fleischer et al. '04, Karakostas-Kolliopoulos,'04 Yang-Huang '04)

For nonatomic routing

Stackelberg routing

- By centrally routing α-fraction of total flow
  - in single-commodity networks: can reduce price of anarchy for any class of latency functions (Roughgarden '03, S '07, …)
  - weaker results known for multicommodity networks
- Given target flow f^* and fraction α, can efficiently find a Stackelberg routing that yields f^* as equilibrium (if one exists)

All algorithmic results:

– equilibrium computation
– finding tolls (to impose a given target flow f^*)
– Stackelberg routing (to impose a given target flow f^*)

assume we have precise, explicit knowledge of latency f'ns

But latency functions may not be known or be unobtainable:

- obtaining detailed information may be costly (time, money)
- may be unable to isolate resources to determine latency f'ns.

Can one analyze routing games without knowing latency f'ns.?

Can we achieve the algorithmic ends—e.g., imposing target flow f^* via tolls/Stackelberg routing—without the means?

Query models

- Know the underlying network and the commodities, but not the latency functions:

$$(G, \cancel{\{\ell_e : \mathbb{R}_+ \to \mathbb{R}_+\}_e}, \{s_i, t_i, d_i\})$$

- Routing game is a black box: can only access via queries
- Efficiency of algorithm measured by:
  - query complexity = no. of queries needed
  - computational complexity

Two types of query models

- Cost/payoff queries

strategy profile
(flow in nonatomic routing)

cost under strategy profile
(total cost, player costs, …)

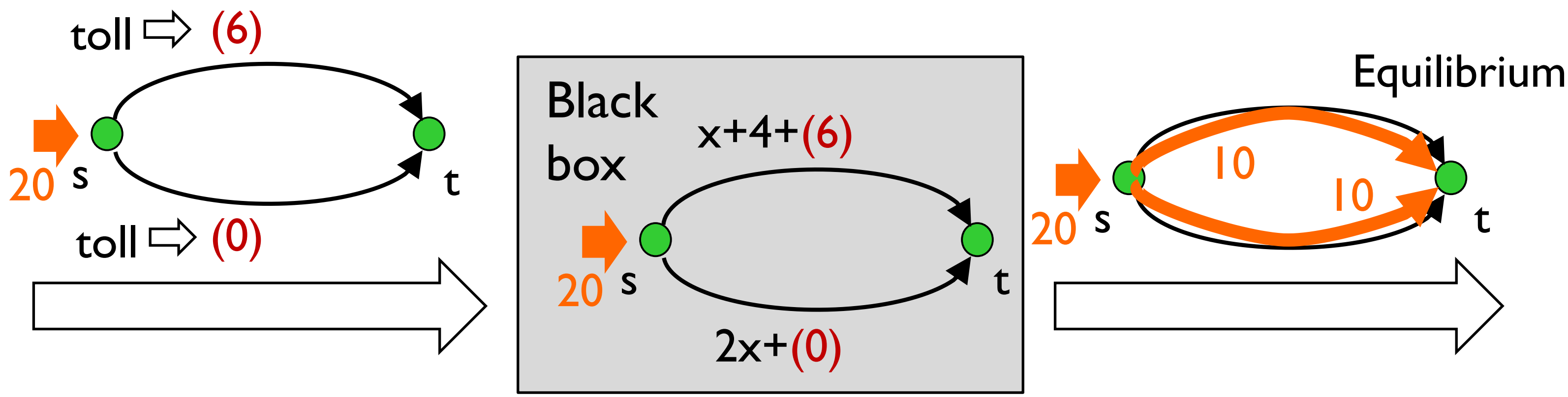
  - Common in empirical game theory, goal: compute equilibria
  - Many variants depending on type of queries and type of equilibria desired (pure/mixed/correlated)
  - Much work for general strategic-form games (Papadimitriou-Roughgarden '08, Hart-Nisan '13, …, work based on regret-dynamics); limited results for routing games (Blum et al. '10, Fisher et al. '06, Kleinberg et al. '09, Fearnley et al. '15; some require info. about unplayed strategies)
  - Criticism: To respond to query, need to route players according to strategy profile to compute cost, but can't dictate routes to players

Two types of query models

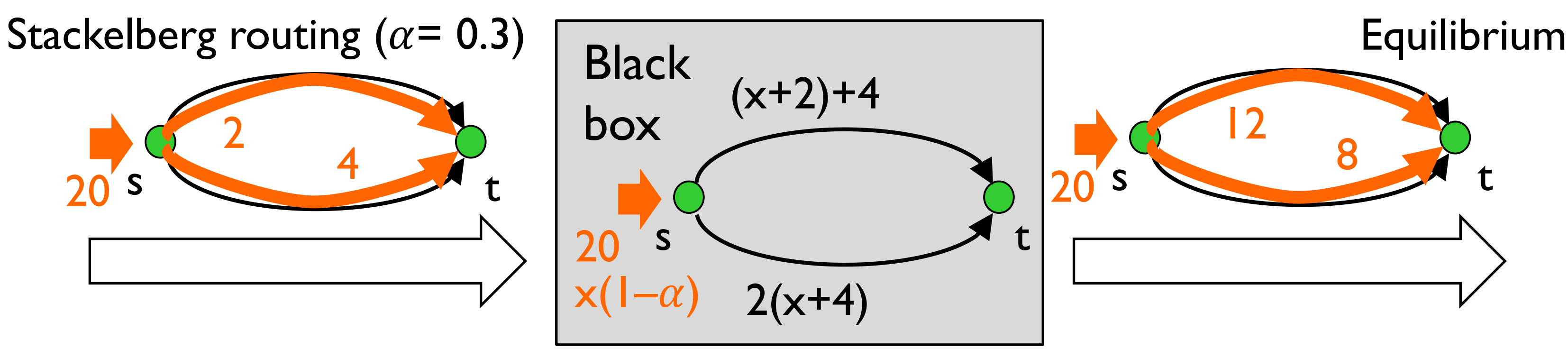
- **Equilibrium queries:** observe equilibrium flow

(Bhaskar-Ligett-Schulman-S '14)

Toll queries



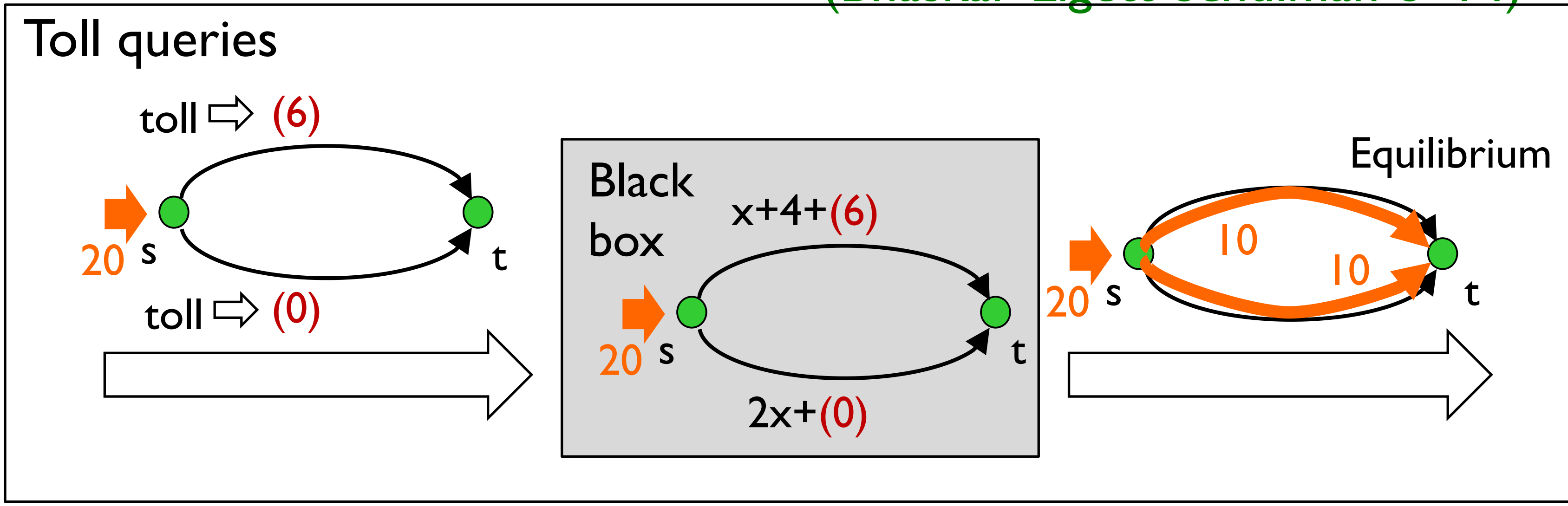
Stackelberg queries



Two types of query models

- Equilibrium queries: observe equilibrium flow

(Bhaskar-Ligett-Schulman-S '14)

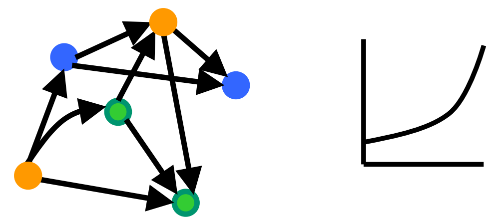


Problem: Given target flow f^* (that is minimal), find tolls $\{\tau_e^*\}_e$ that yield f^* as equilibrium flow using polynomial no. of toll queries (and preferably, polytime computation)

Results (Bhaskar-Ligett-Schulman-S '14)

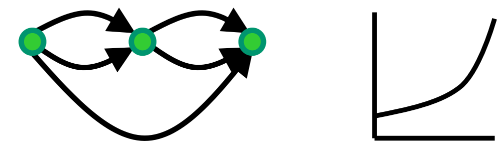
Polynomial query complexity for general graphs, general (polynomial) latency f'ns.

- novel application of the ellipsoid method

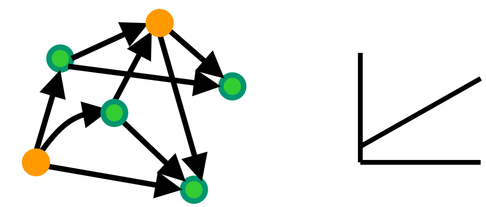


Improved query-complexity bounds for

- series-parallel graphs, general latency f'ns.



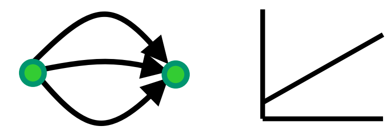
- general single-commodity networks, linear latency functions



All algorithms are polytime; also, with non-linear latencies, only require that black box returns approximate equilibria (bounds only meaningful under this relaxation as equilibria can be irrational)

Results: lower bounds (BLSS '14)

Need $\geq |E| - 1$ queries, even for parallel links, linear latency functions



Can one learn ~~the latency functions~~? equivalent latency f'ns.?

Latency f'ns. $\{\ell_e\}_e,\ \{\ell'_e\}_e$ are (toll-) equivalent $\Leftrightarrow$ they yield same equilibrium for all edge tolls

Q'n: Can one use toll queries to obtain $\{\ell'_e\}_e$ that are equivalent to actual latency f'ns $\{\ell_e\}_e$?

OPEN! Seems difficult (at least with poly-many queries)

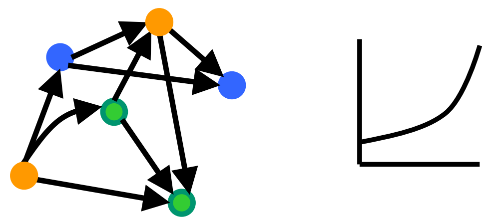
Computational q'n: Given $\{\ell_e\}_e,\ \{\ell'_e\}_e$, determine if they are not equivalent.

NP-hard (even if each ℓ_e, ℓ'_e is const.)

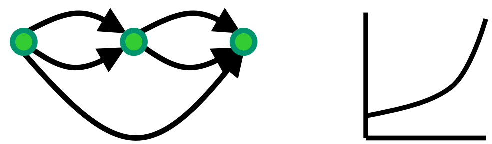
Our algorithms are doing something less taxing than learning latency f'ns. – learning "just enough" to impose target flow

Results (Bhaskar-Ligett-Schulman-S '14)

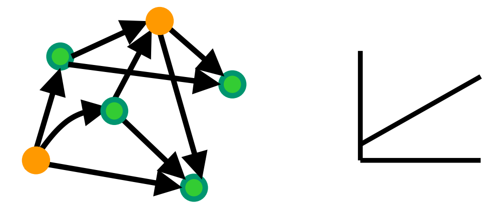
Polynomial query complexity for general graphs, general (polynomial) latency f'ns.
- novel application of the ellipsoid method



Improved query-complexity bounds for
- series-parallel graphs, general latency f'ns.



- general single-commodity networks, linear latency functions



All algorithms are polytime; also, with non-linear latencies, only require that black box returns approximate equilibria (bounds only meaningful under this relaxation as equilibria can be irrational)

Enforcing target flow via toll queries

Given: target flow f^* (assume is minimal),
toll queries for nonatomic routing game

This talk: (i) single commodity (minimal $\equiv$ acyclic)
(ii) linear latency f'ns. $a_e^* x + b_e^*$ on each edge e

Let $\{\tau_e^*\}_e$ be tolls that impose f^*

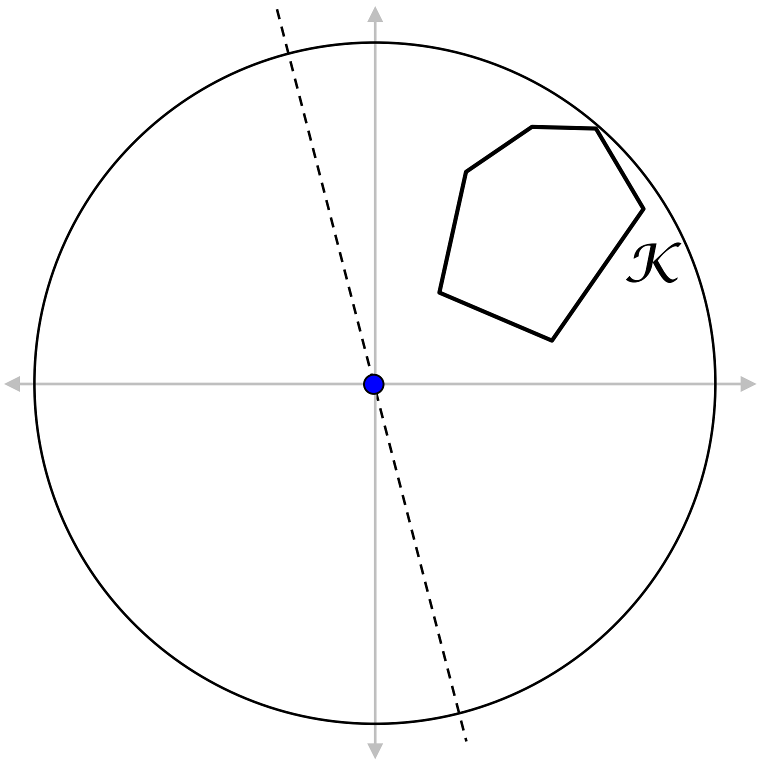
(Recall: Tolls τ^* always exist (since f^* is minimal)

f is equilibrium if whenever $f_e > 0\ \forall e \in s$-t path P, we have $\sum_{e \in P} \ell_e(f_e) \leq \sum_{e \in Q} \ell_e(f_e)$ for all s-t paths Q)

IDEA: Use ellipsoid method to search for the point $(a_e^*, b_e^*, \tau_e^*)_e$

The Ellipsoid Method

$\mathcal{K} \subseteq \mathbb{R}^n$ Find $x \in \mathcal{K}$, or determine $\mathcal{K} = \emptyset$



Ellipsoid ≡ squashed sphere

Start with ball of radius R containing $\mathcal{K}$.

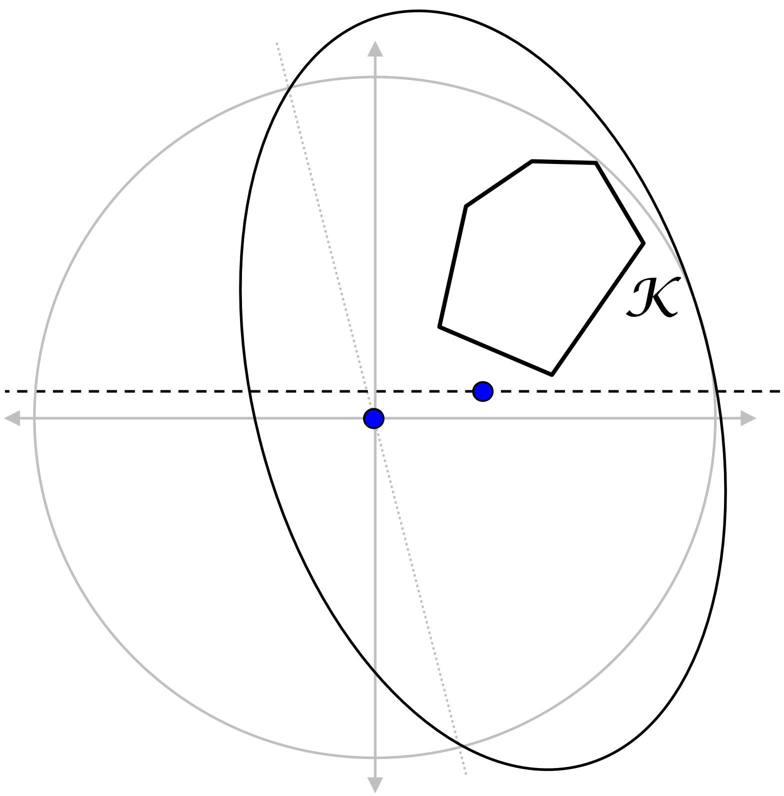
y_i = center of current ellipsoid.

If $y_i \notin \mathcal{K}$, find violated inequality $a \cdot x \leq a \cdot y_i$ to chop off infeasible half-ellipsoid.

Separation oracle

The Ellipsoid Method

$\mathcal{K} \subseteq \mathbb{R}^n$ Find $x \in \mathcal{K}$, or determine $\mathcal{K} = \emptyset$



Ellipsoid ≡ squashed sphere

Start with ball of radius R containing $\mathcal{K}$.

y_i = center of current ellipsoid.

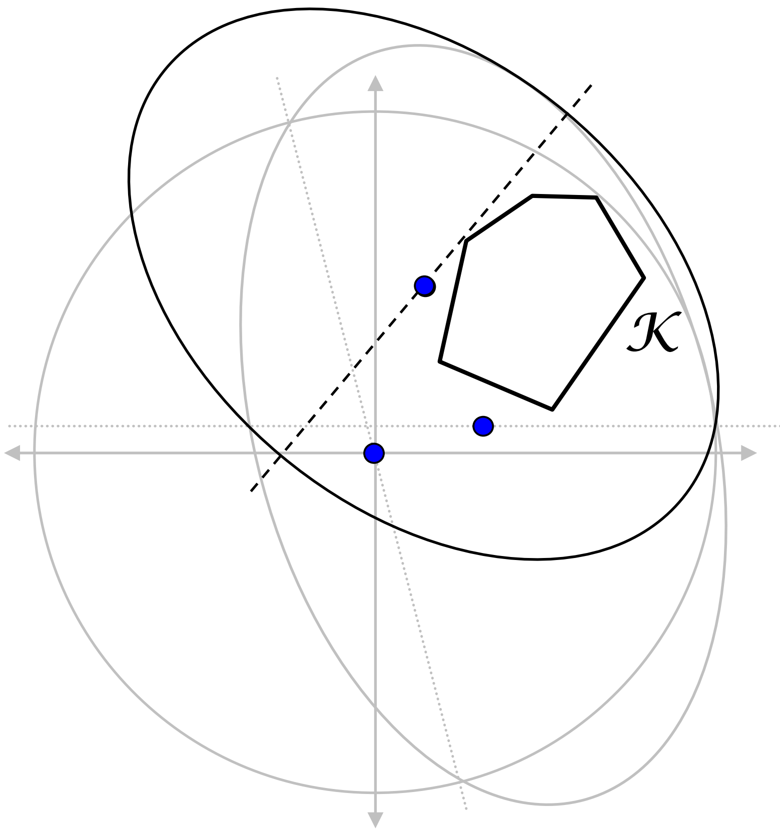
If $y_i \notin \mathcal{K}$, find violated inequality $a \cdot x \leq a \cdot y_i$ to chop off infeasible half-ellipsoid.

New ellipsoid = min. volume ellipsoid containing "unchopped" half-ellipsoid.

Repeat for $i=0,1,\ldots,T$

The Ellipsoid Method

$\mathcal{K} \subseteq \mathbb{R}^n$ Find $x \in \mathcal{K}$, or determine $\mathcal{K} = \emptyset$



Ellipsoid ≡ squashed sphere

Start with ball of radius R containing $\mathcal{K}$.

y_i = center of current ellipsoid.

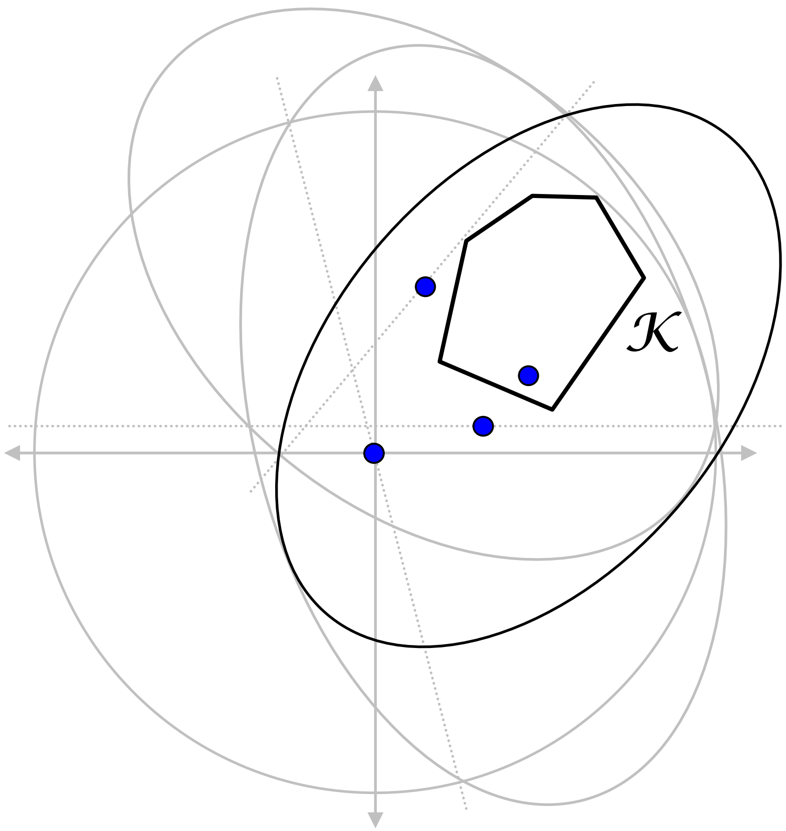
If $y_i \notin \mathcal{K}$, find violated inequality $a \cdot x \leq a \cdot y_i$ to chop off infeasible half-ellipsoid.

New ellipsoid = min. volume ellipsoid containing "unchopped" half-ellipsoid.

Repeat for $i=0,1,\ldots,T$

The Ellipsoid Method

$\mathcal{K} \subseteq \mathbb{R}^n$ Find $x \in \mathcal{K}$, or determine $\mathcal{K} = \emptyset$



Ellipsoid ≡ squashed sphere

Start with ball of radius R containing $\mathcal{K}$.

y_i = center of current ellipsoid.

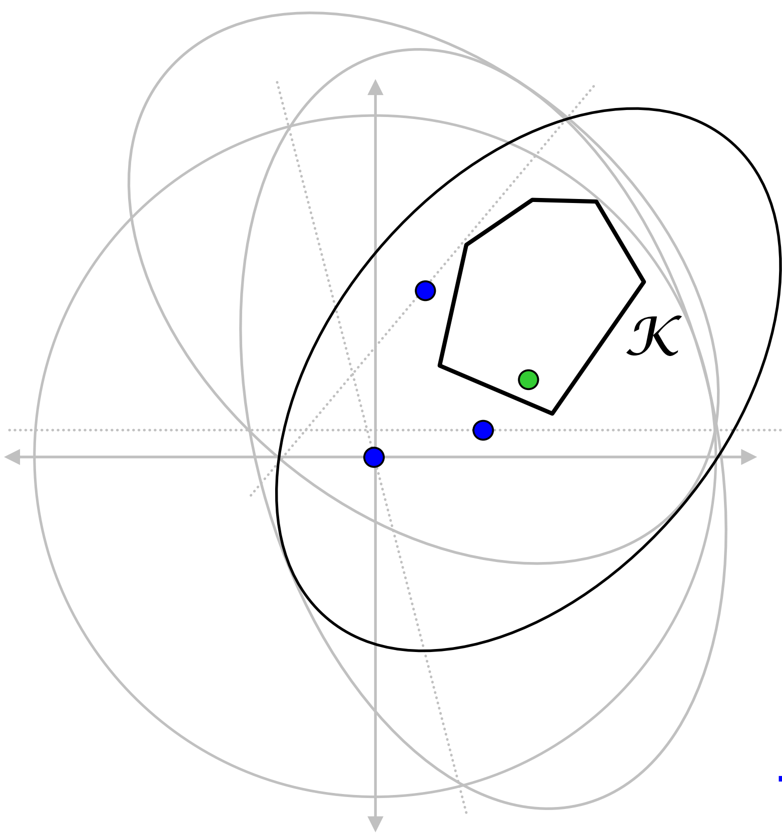
If $y_i \notin \mathcal{K}$, find violated inequality $a \cdot x \leq a \cdot y_i$ to chop off infeasible half-ellipsoid.

New ellipsoid = min. volume ellipsoid containing "unchopped" half-ellipsoid.

Repeat for $i=0,1,\ldots,T$

The Ellipsoid Method

$\mathcal{K} \subseteq \mathbb{R}^n$ Find $x \in \mathcal{K}$, or determine $\mathcal{K} = \emptyset$



Ellipsoid ≡ squashed sphere

Start with ball of radius R containing $\mathcal{K}$.

y_i = center of current ellipsoid.

If $y_i \notin \mathcal{K}$, find violated inequality $a \cdot x \le a \cdot y_i$ to chop off infeasible half-ellipsoid.

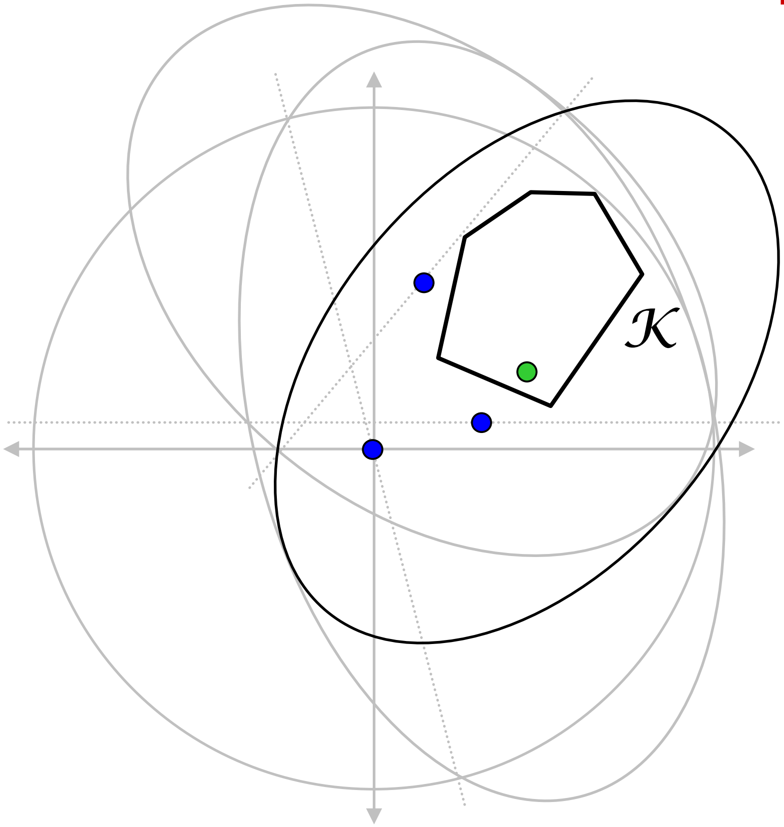
If $y_i \in \mathcal{K}$, Done!

New ellipsoid = min. volume ellipsoid containing "unchopped" half-ellipsoid.

Repeat for $i=0,1,\ldots,T$

$$T = \mathrm{poly}\left(n, \ln\left(\frac{R}{\text{radius of ball contained in } \mathcal{K}}\right)\right)$$

The Ellipsoid Method



Start with ball of radius R containing $\mathcal{K}$.

y_i = center of current ellipsoid.

If $y_i \notin \mathcal{K}$, find violated inequality $a \cdot x \leq a \cdot y_i$ to chop off infeasible half-ellipsoid.

If $y_i \in \mathcal{K}$, Done!

New ellipsoid = min. volume ellipsoid containing "unchopped" half-ellipsoid.

Rep

maximum bit complexity of vertex or facet of $\mathcal{K}$

Theorem (Grotschel-Lovasz-Schrijver):

$\mathcal{K} \subseteq \mathbb{R}^n$: polytope of encoding size M

have separation oracle that if $y \notin \mathcal{K}$ returns hyperplane of size $\leq$ size(y), M

Can use ellipsoid method to find $x \in \mathcal{K}$, or determine $\mathcal{K} = \emptyset$, in polytime, using poly(n, M) calls to separation oracle

Enforcing target flow via toll queries

Given: target flow f^* (assume is minimal),
toll queries for nonatomic routing game

This talk: (i) single commodity (minimal $\equiv$ acyclic)
(ii) linear latency f'ns. $a_e^* x + b_e^*$ on each edge e

Let $\{\tau_e^*\}_e$ be tolls that impose f^*

IDEA: Use ellipsoid method to search for the point $(a_e^*, b_e^*, \tau_e^*)_e$

Take $\mathcal{K} = \{(a_e^*, b_e^*, \tau_e^*)_e\}$ → singleton set!

Encoding length = bit size of $(a_e^*, b_e^*, \tau_e^*)_e$ = M (part of input)

Show: given center $p = \left(\hat{a}_e, \hat{b}_e, \hat{\tau}_e\right)_e$ of current ellipsoid,
tolls $\hat{\tau}$ do not yield f^* $\Longrightarrow$ can find hyperplane separating p from $\mathcal{K}$

Enforcing target flow via toll queries

Take $\mathcal{K} = \{(a_e^*, b_e^*, \tau_e^*)_e\}$ → singleton set!

Show: given center $p = \left(\hat{a}_e, \hat{b}_e, \hat{\tau}_e\right)_e$ of current ellipsoid,

tolls $\hat{\tau}$ do not yield f^* $\Longrightarrow$ can find hyperplane separating p from $\mathcal{K}$

1) $f^* \neq$ equilibrium flow for latency f'ns. $\left\{\hat{a}_e x + \hat{b}_e\right\}_e$, tolls $\{\hat{\tau}_e\}_e$

Then $\exists$ **s-t** paths P, Q (can be found efficiently) s.t. $f_e^* > 0 \;\; \forall e \in P$,

but $\sum_{e\in P}\left(\hat{a}_e f_e^* + \hat{b}_e + \hat{\tau}_e\right) > \sum_{e\in Q}\left(\hat{a}_e f_e^* + \hat{b}_e + \hat{\tau}_e\right)$

Also f^* = equilibrium flow for latency f'ns. $\{a_e^* x + b_e^*\}_e$, tolls $\{\tau_e^*\}_e$

So, $\sum_{e\in P}(a_e^* f_e^* + b_e^* + \tau_e^*) \leq \sum_{e\in Q}(a_e^* f_e^* + b_e^* + \tau_e^*)$

Then $\sum_{e\in P}\left(a_e\, f_e^* + b_e \;\; + \tau_e \;\right) \leq \sum_{e\in Q}\left(a_e\, f_e^* + b_e \;\; + \tau_e \;\right)$

is an inequality violated by $\left(\hat{a}_e, \hat{b}_e, \hat{\tau}_e\right)_e$, but satisfied by $\mathcal{K}$

Enforcing target flow via toll queries

Take $\mathcal{K} = \{(a_e^*, b_e^*, \tau_e^*)_e\}$ → singleton set!

Show: given center $p = \left(\hat{a}_e, \hat{b}_e, \hat{\tau}_e\right)_e$ of current ellipsoid,

tolls $\hat{\tau}$ do not yield $f^* \Longrightarrow$ can find hyperplane separating p from $\mathcal{K}$

1) If $f^* \neq$ equilibrium flow for latency f'ns. $\left\{\hat{a}_e x + \hat{b}_e\right\}_e$, tolls $\{\hat{\tau}_e\}_e$ ✓

2) So let f^* = equilibrium flow for latency f'ns. $\left\{\hat{a}_e x + \hat{b}_e\right\}_e$, tolls $\hat{\tau}$

Let f = equilibrium flow for latency f'ns. $\{a_e^* x + b_e^*\}_e$, tolls $\hat{\tau}$

(obtain from black box)

Enforcing target flow via toll queries

Take $\mathcal{K} = \{(a_e^*, b_e^*, \tau_e^*)_e\}$ → singleton set!

Show: given center $p = (\hat{a}_e, \hat{b}_e, \hat{\tau}_e)_e$ of current ellipsoid,

tolls $\hat{\tau}$ do not yield f^* $\Longrightarrow$ can find hyperplane separating p from $\mathcal{K}$

1) If $f^* \neq$ equilibrium flow for latency f'ns. $\{\hat{a}_e x + \hat{b}_e\}_e$, tolls $\{\hat{\tau}_e\}_e$ ✓

2) So let f^* = equilibrium flow for latency f'ns. $\{\hat{a}_e x + \hat{b}_e\}_e$, tolls $\hat{\tau}$

Let f = equilibrium flow for latency f'ns. $\{a_e^* x + b_e^*\}_e$, tolls $\hat{\tau}$

$f \neq f^*$, so $f \neq$ equilibrium flow for latency f'ns. $\{\hat{a}_e x + \hat{b}_e\}_e$, tolls $\hat{\tau}$

Again $\exists$ s-t paths P, Q (can be found efficiently) s.t. $f_e > 0 \;\forall e \in P$,

$$\sum_{e \in P}(\hat{a}_e f_e + \hat{b}_e + \hat{\tau}_e) > \sum_{e \in Q}(\hat{a}_e f_e + \hat{b}_e + \hat{\tau}_e)$$

but $$\sum_{e \in P}(a_e^* f_e + b_e^* + \hat{\tau}_e) \leq \sum_{e \in Q}(a_e^* f_e + b_e^* + \hat{\tau}_e)$$

Then $$\sum_{e \in P}(a_e f_e + b_e + \hat{\tau}_e) \leq \sum_{e \in Q}(a_e f_e + b_e + \hat{\tau}_e)$$

is an inequality violated by $(\hat{a}_e, \hat{b}_e, \hat{\tau}_e)_e$, but satisfied by $\mathcal{K}$

Enforcing target flow via toll queries

Theorem (BLSS '14): Using polynomial no. of toll queries, can find tolls that enforce f^*, or deduce that no such tolls exist, for:

- general nonatomic routing games (general graphs, latency f'ns.)
- nonatomic routing with linear constraints on tolls
  - E.g., disallowing tolls, or bounding total toll paid by player
- nonatomic congestion games

(Roth et al. '16 also obtain some of the above results using different methods.)

Improved bounds for:

- series-parallel graphs, general latency functions
- general single-commodity networks, linear latency functions

obtained by deriving new properties of tolls, multicommodity flows in series-parallel graphs, and sensitivity of equilibria to tolls

Open directions with toll queries

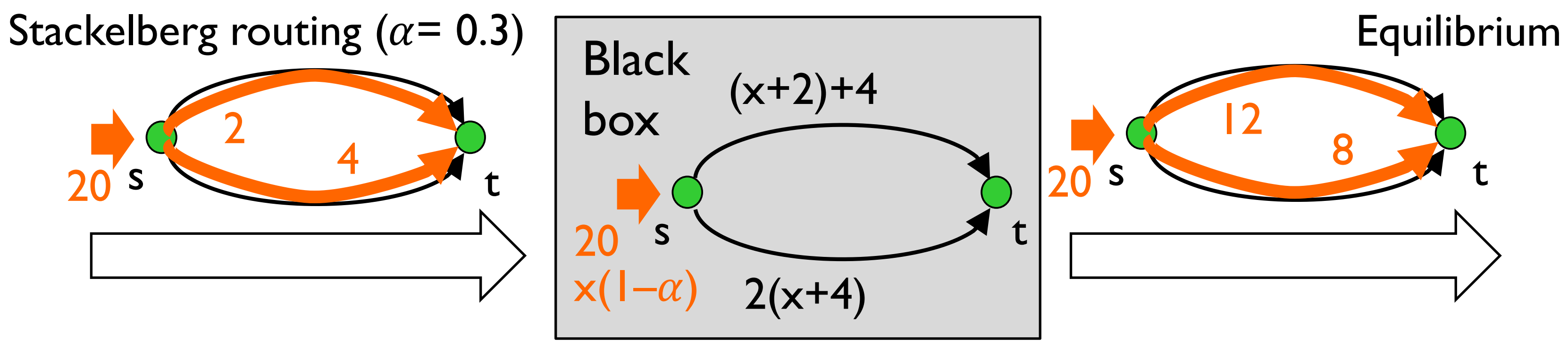
- What about atomic routing games?
  - Quite open, for both unsplittable and splittable routing

    (RECALL: players now control finite amounts of demand, choose how to route their demand unsplittably/splittably from their source to sink)
  - If we assume equilibria are unique for all latency f'ns. encountered during ellipsoid, then machinery extends
  - Challenge: get rid of uniqueness assumption
  - Other issues:
    - do not understand what target flows can be induced (uniquely)
    - for atomic unsplittable routing, pure equilibria need not exist – useful to focus first on settings where equilibria always exist (e.g., uniform demands and/or linear latencies)

Open directions with toll queries

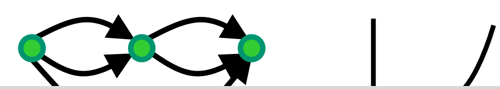
- What about atomic routing games?
  - Quite open, for both unsplittable and splittable routing
    (RECALL: players now control finite amounts of demand, choose how to route their demand unsplittably/splittably from their source to sink)
  - If we assume equilibria are unique for all latency f'ns. encountered during ellipsoid, then machinery extends
  - Challenge: get rid of uniqueness assumption
- Better upper/lower bounds on query complexity?
- What if we are allowed only a given fixed no. of queries? Or making query incurs cost, and have a budget on total query cost?
  - Can we obtain flow $f(k)$ after k queries such that distance between $f(k)$ and f^* decreases (nicely) with k?

Stackelberg queries



Problem: Given target flow f^* and α, find Stackelberg routing that yields f^* as equilibrium using polynomial no. of Stackelberg queries (focus on single-commodity networks)

BLSS '14: solve problem for series-parallel graphs



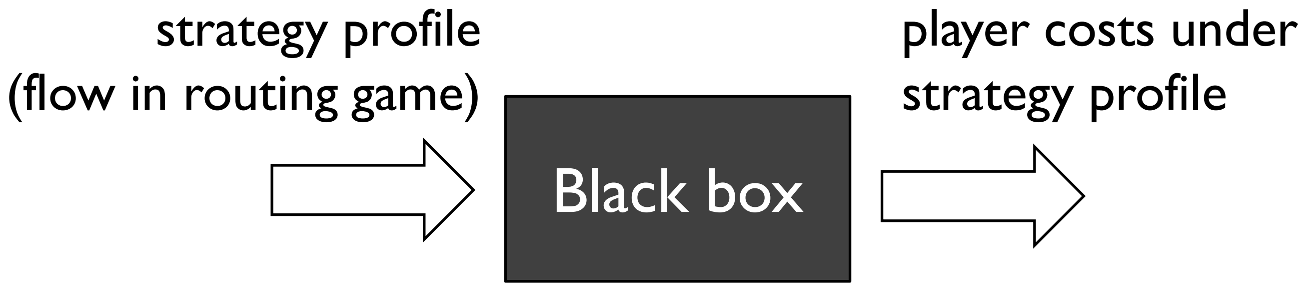
latency f'ns. $\{\ell_e\}_e, \{\ell'_e\}_e$ are Stackelberg-equivalent $\Leftrightarrow$ they yield same equilibrium for all Stackelberg routings

Everything else i

BLSS '14: learning latency f'ns. that are Stackelberg-equivalent to true latency f'ns. requires exponential no. of queries

– also NP-hard when latency f'ns. are explicitly given

Cost queries: equilibrium computation



Nonatomic routing: algorithms by Blum et al. '10, Fisher et al. '06

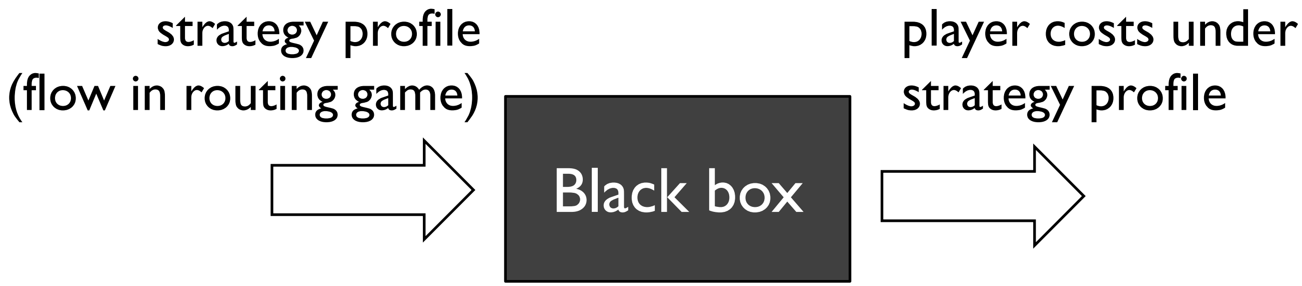
Atomic splittable routing: equilibrium computation not well-understood even when latency f'ns. are explicitly given

Focus on atomic unsplittable routing & computing pure Nash equilibrium

NOT MUCH IS KNOWN

- Kleinberg et al. '09: require knowledge also about unplayed strategies
- Fearnley et al. '15: obtain results for single source-sink parallel-link graphs and single source-sink DAGs
- Challenge in adapting online learning results: get information about costs, but equilibrium involves minimizing a different potential function

Cost queries: equilibrium computation



Focus on atomic unsplittable routing & computing pure Nash equilibrium

NOT MUCH IS KNOWN

Start really simple: single source-sink pair, only 1 player

related to graph discovery, network tomography

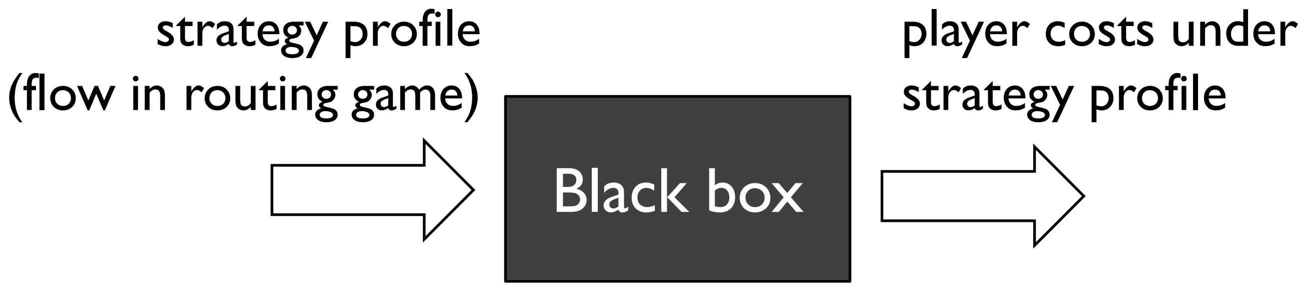
I.e., compute s-t shortest path using path-cost queries (edge costs ≥ 0)

O(|E|) queries suffice (joint work with Bhaskar, Gairing, Savani)

0 queries

- Find set B $\subseteq \mathcal{P}$:= {simple s-t paths} s.t. aff-span(B) contains $\mathcal{P}$
- Query costs of all paths in B
- Solve LP: minimize cost(f) s.t. f ∈ aff-span(B), $f \geq 0$.
- Decompose f into simple s-t paths, cycles; one of the paths is shortest s-t path

Cost queries: equilibrium computation



Focus on atomic unsplittable routing & computing pure Nash equilibrium

NOT MUCH IS KNOWN

Start ^really^ simple: single source-sink pair, only 1 player

related to graph discovery, network tomography

I.e., compute s-t shortest path using path-cost queries (edge costs ≥ 0)

O(|E|) queries suffice (joint work with Bhaskar, Gairing, Savani)

- Find set B ⊆ $\mathcal{P}$:= {simple s-t paths} s.t. aff-span(B) contains $\mathcal{P}$

0 queries coNP-hard

OPEN: algorithm with polynomial query- and time- complexity?

(and more generally, for computing NE for unsplittable routing)

Summary

- Query models: new perspective on routing games
  - Do not assume latency functions are explicitly given
  - Black-box access to routing games via queries
- Present various new challenges
- Various models
  - Cost queries (input: strategy profile, output: player costs)
  - Toll queries (input: tolls, output: equilibrium flow)
  - Stackelberg queries (input: Stackelberg routing, output: equilibrium)
  - Can consider other models: best/better-response queries
- Strongest results known are for nonatomic games with cost queries and toll queries
- Atomic routing games: many gaps, don't understand well
  - Even "simple" special cases pose interesting open questions

Thank you

Any queries?