dispatch

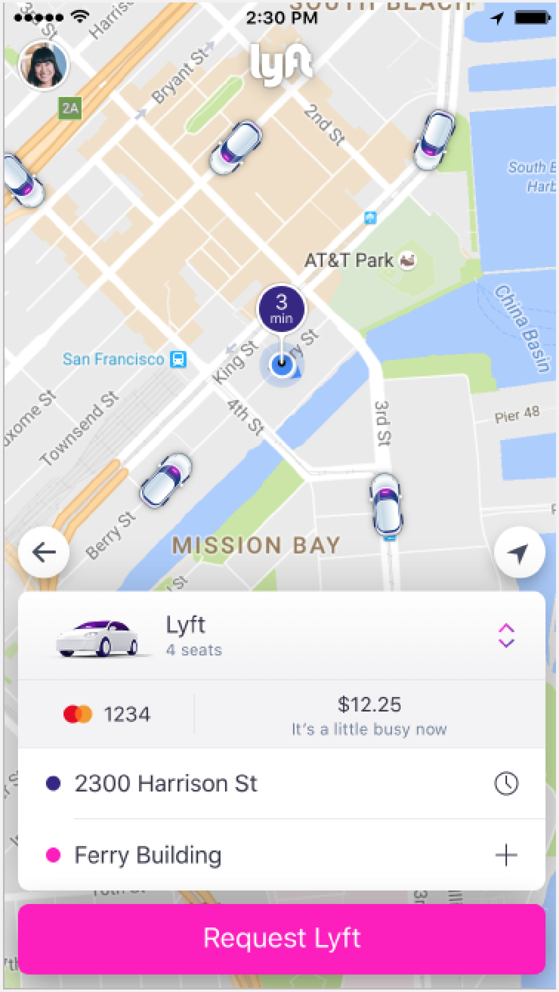
Sid Banerjee
School of ORIE, Cornell University




joint work with – Daniel Freund, Thodoris Lykouris (Cornell),
– Pengyu Qian & Yash Kanoria (Columbia)

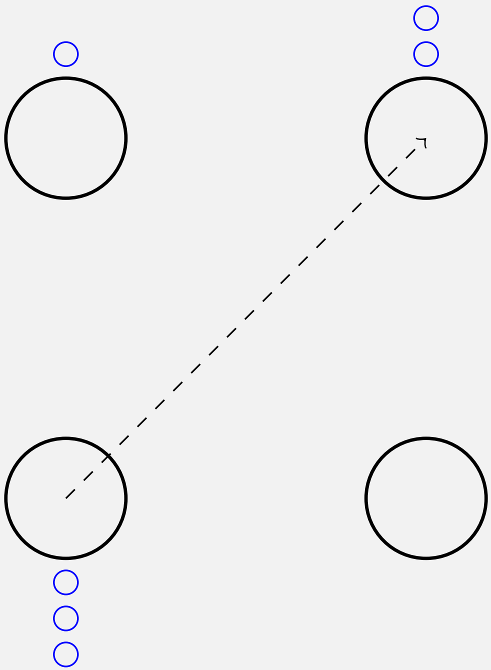
work supported by ARO grant W911NF- 17-1-0094.

the dispatch problem



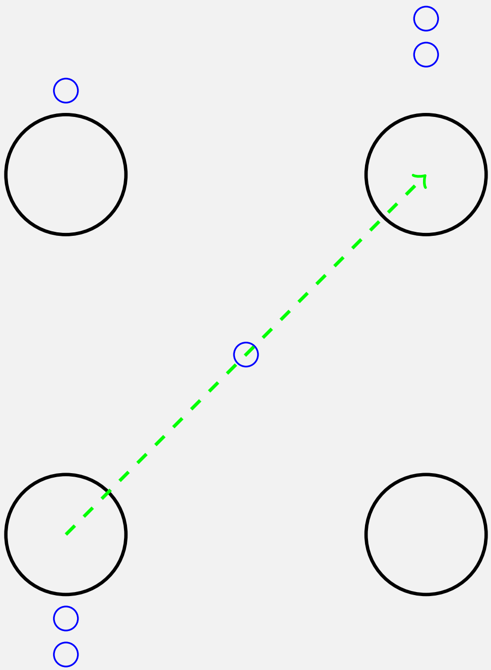
- input: ride request (source, destn)
- output: match to 'nearby' car
- aim: minimize missed requests

(stochastic) model for ridesharing



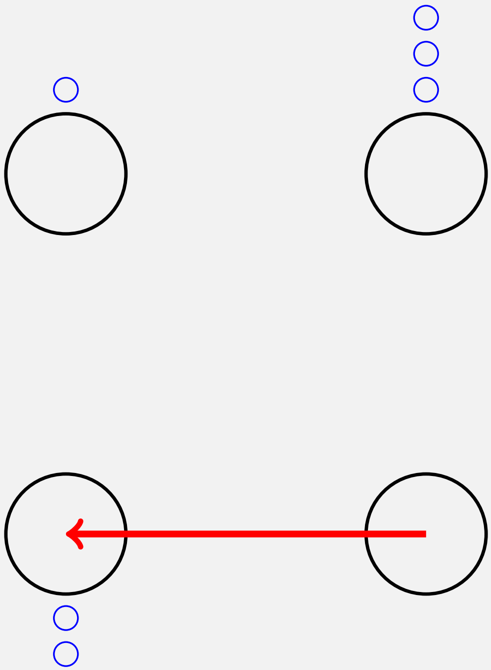
- K units (cars) across n stations (closed network)
- system state $\in S_{n,K} = \{(x_i)_{i\in[n]} | \sum_{i=1}^{n} x_i = K\}$
- $i \to j$ passengers arrive via Poisson process with rate ϕ_{ij}

(stochastic) model for ridesharing



- passenger requests ride if offered price is acceptable
- matched to idle unit, which then travels to destination
- trips have independent travel-times (for this talk, trip-times = zero)

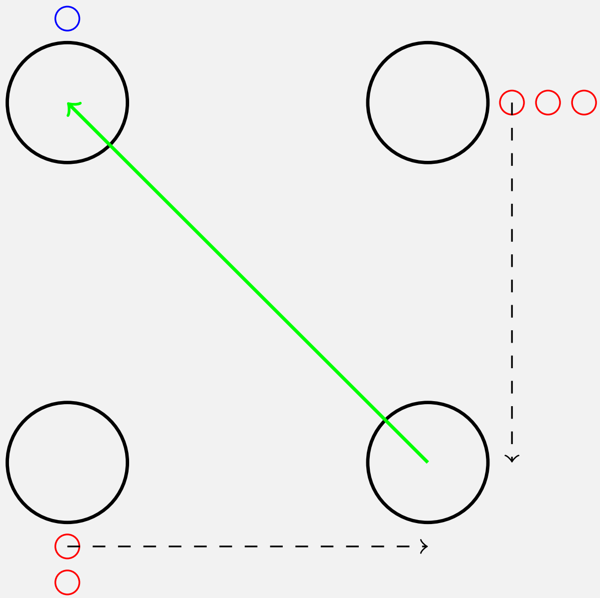
(stochastic) model for ridesharing



- myopic customers: abandon system if unit unavailable

aim: minimize long-term average rate of missed requests (w/ fixed prices)

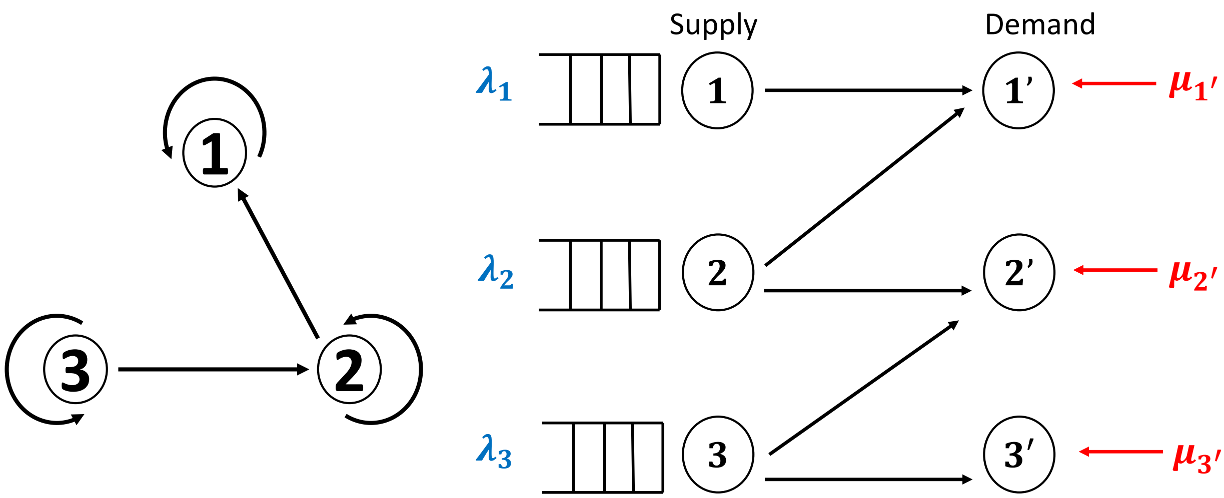
main control lever for this talk



- dispatch: choose 'nearby' car to serve demand

assumption: allowed to use any unit within given 'ETA target'

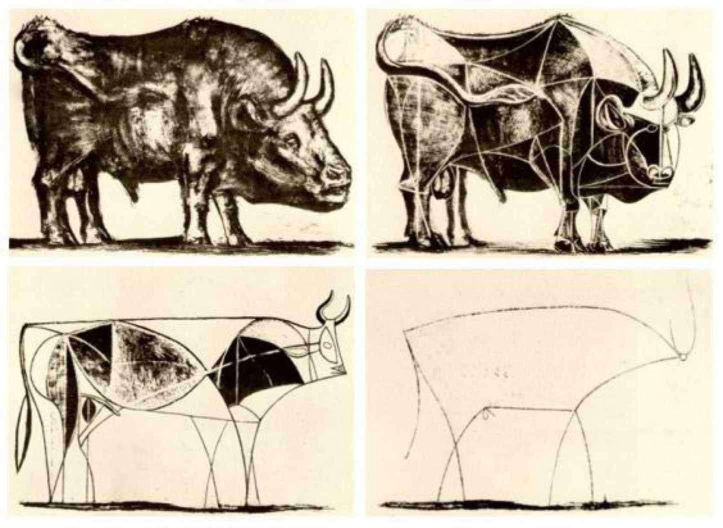
bipartite matching with externalities



$$\lambda_i \triangleq \sum_{k'=1'}^{n'} \phi_{k'i}\,, \quad \mu_{j'} \triangleq \sum_{i=1}^{n} \phi_{j'i}\,, \qquad \text{Note } \sum_{i \in V} \lambda_i = \sum_{j' \in V'} \mu_{j'}\,.$$

- to meet all demand at $1'$, we need $\lambda_1 + \lambda_2 \geq \mu_{1'}$
- how well can we handle stochastic fluctuations?

intermezzo: the many schools of modeling



all models are innacurate; the question drives the model

what do we want to answer?

- what is the correct structure of dispatch policies?
- what is the value of state-dependent control ('real-time')?
- what is the role of information?
- can we separate pricing and dispatch?

intermezzo: the joy of theory

connections to deep theoretical questions in algorithms and control

two amazing recent theoretical advances

- optimal control of input-queued switches (Maguluri & Srikant 2016)
- the k-server conjecture (Bubeck, Cohen, Lee, Lee, Madry 2017)

intriguing connections between the approaches. . .

formal problem

- n stations, K units, Poisson(ϕ_{jk}) demand arrivals
  - connectedness: $[\phi_{jk}]$ is irreducible.
  - non-triviality: $\exists(j,k)$ such that $\phi_{jk} > 0$ and $(k,j) \notin E$
- compatibility E: $(i,j) \in E \Rightarrow$ supply at i can serve demand at j
  - self-compatibility: $(j,j) \in E$ for all $j \in V$

objective

$$\max_{\text{dispatch rules } \vec{q}} \underbrace{\sum_{\mathbf{x}} \pi_{\vec{q}}(\mathbf{x})}_{\text{long-run avg under } \vec{q}} \left(\sum_{e=(i,j)} \phi_{ij} \underbrace{\chi_{ij}(q_{ij}(\mathbf{x}))}_{ij \text{ request served under } q(\mathbf{x})} \right)$$

formal problem

- n stations, K units, Poisson(ϕ_{jk}) demand arrivals
- compatibility E: $(i, j) \in E \Rightarrow$ supply at i can serve demand at j

objective

$$\max_{\vec{q}} \mathbb{E}_{\pi_{\vec{q}}(\mathbf{X})} \left[\sum_{ij} \phi_{ij} \chi_{ij}(q_{ij}(\mathbf{X})) \right]$$

challenges

- exponential size of policy

formal problem

- n stations, K units, Poisson(ϕ_{jk}) demand arrivals
- compatibility E: $(i,j) \in E \Rightarrow$ supply at i can serve demand at j

objective

$$\max_{\vec{q}} \mathbb{E}_{\pi_{\vec{q}}(\mathbf{X})} \left[\sum_{ij} \phi_{ij} \chi_{ij}(q_{ij}(\mathbf{X})) \right]$$

challenges

- exponential size of policy
- non-convex problem: even with state-independent $\vec{q}$

what can we show

$$\max_{\vec{q}} \mathbb{E}_{\pi_{\vec{q}}(\mathbf{X})}\left[\sum_{ij} \phi_{ij}\chi_{ij}(q_{ij}(\mathbf{X}))\right]$$

theorem [B, Freund & Lykouris 2017]

flow relaxation gives state-independent dispatch policy which is

- $1 + \frac{n-1}{K}$ approximate (with instantaneous trips)
- $1 + O\left(\frac{1}{\sqrt{K}}\right)$ approximate (with travel-times)

what can we show

$$\max_{\vec{q}} \mathbb{E}_{\pi_{\vec{q}}(\mathbf{X})} \left[\sum_{ij} \phi_{ij} \chi_{ij}(q_{ij}(\mathbf{X})) \right]$$

theorem [B, Freund & Lykouris 2017]

flow relaxation gives state-independent dispatch policy which is

- $1 + \frac{n-1}{K}$ approximate (with instantaneous trips)
- $1 + O\left(\frac{1}{\sqrt{K}}\right)$ approximate (with travel-times)

- extends to pricing, rebalancing controls, most objectives
- large-supply/large-market optimality: factor goes to 1 as system scales

what can we show

$$\max_{\vec{q}} \mathbb{E}_{\pi_{\vec{q}}(\mathbf{X})} \left[\sum_{ij} \phi_{ij} \chi_{ij}(q_{ij}(\mathbf{X})) \right]$$

theorem [B, Freund & Lykouris 2017]

flow relaxation gives state-independent dispatch policy which is

- $1 + \frac{n-1}{K}$ approximate (with instantaneous trips)
- $1 + O\left(\frac{1}{\sqrt{K}}\right)$ approximate (with travel-times)

theorem [B, Kanoria & Qian 2018]

family of state-dependent dispatch policies which are

- $1 + e^{-\Theta(K)}$ approximate (for large K, instantaneous trips)
- convex program gives optimal exponent

state-independent dispatch: proof roadmap

relaxation + resource augmentation

step 1: elevated flow relaxation (EFR): relax objective into a flow program

- optimize over fraction of $i \to j$ requests accepted
- encode conservation laws: flow balance, Little's law

state-independent dispatch: proof roadmap

relaxation + resource augmentation

step 1: elevated flow relaxation (EFR): relax objective into a flow program

- optimize over fraction of $i \to j$ requests accepted
- encode conservation laws: flow balance, Little's law

step 2: show EFR is tight under state-independent dispatch policies, in the 'infinite-unit system' (i.e., $K \to \infty$)

state-independent dispatch: proof roadmap

relaxation + resource augmentation

step 1: elevated flow relaxation (EFR): relax objective into a flow program

- optimize over fraction of $i \to j$ requests accepted
- encode conservation laws: flow balance, Little's law

step 2: show EFR is tight under state-independent dispatch policies, in the 'infinite-unit system' (i.e., $K \to \infty$)

step 3: construct sequence of product-form Markov chains for finite K based on the optimal infinite-unit policy

- bound the normalization constant for these chains to get result

in summary

theorem [B, Freund & Lykouris 2017]

state-independent control policies $\vec{q}_\infty$ (from EFR) in K-unit system gives

$$OBJ_K(\vec{q}_\infty) \geq \alpha_{Kn} OPT_K \quad , \quad \text{where } \alpha_{Kn} = \frac{K}{K+n-1}$$

in summary

theorem [B, Freund & Lykouris 2017]

state-independent control policies $\vec{q}_\infty$ (from EFR) in K-unit system gives

$$OBJ_K(\vec{q}_\infty) \geq \alpha_{Kn} OPT_K \quad , \quad \text{where } \alpha_{Kn} = \frac{K}{K+n-1}$$

main takeaways

new techniques for state-independent control of closed networks

- extends to more complex settings
  (travel-times, multi-objective, pooling, reservations)

in summary

theorem [B, Freund & Lykouris 2017]

state-independent control policies $\vec{q}_\infty$ (from EFR) in K-unit system gives

$$OBJ_K(\vec{q}_\infty) \geq \alpha_{Kn} OPT_K \quad , \quad \text{where } \alpha_{Kn} = \frac{K}{K+n-1}$$

main takeaways

new techniques for state-independent control of closed networks

- extends to more complex settings
  (travel-times, multi-objective, pooling, reservations)
- needs demand-rate and price-elasticity estimates
- guarantees are tight for state-independent dispatch policies

in summary

theorem [B, Freund & Lykouris 2017]

state-independent control policies $\vec{q}_\infty$ (from EFR) in K-unit system gives

$$OBJ_K(\vec{q}_\infty) \geq \alpha_{Kn} OPT_K \quad , \quad \text{where } \alpha_{Kn} = \frac{K}{K+n-1}$$

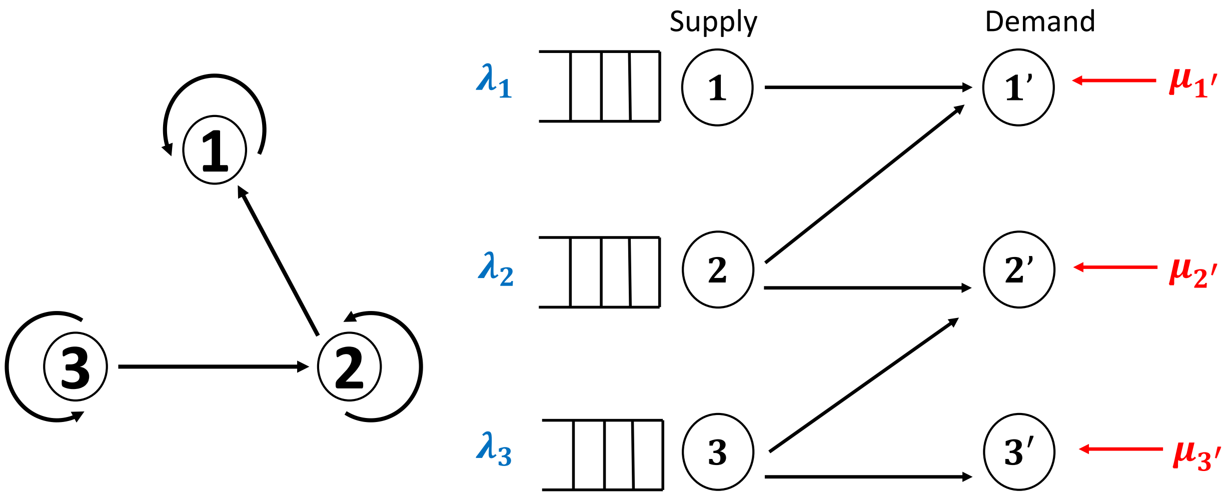
main takeaways

new techniques for state-independent control of closed networks

- extends to more complex settings (travel-times, multi-objective, pooling, reservations)
- needs demand-rate and price-elasticity estimates
- guarantees are tight for state-independent dispatch policies

how do we go beyond this?

model: bipartite matching with externalities

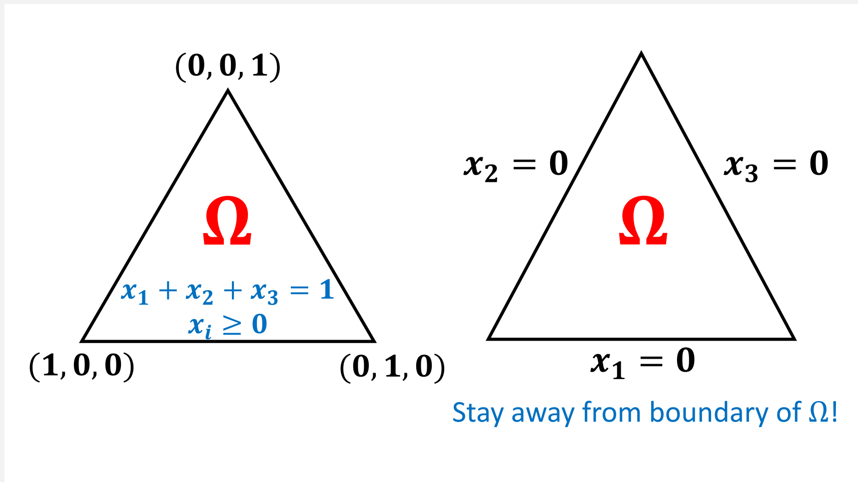


$$\lambda_i \triangleq \sum_{k'=1'}^{n'} \phi_{k'i} \,, \quad \mu_{j'} \triangleq \sum_{i=1}^{n} \phi_{j'i} \,, \qquad \text{Note} \sum_{i \in V} \lambda_i = \sum_{j' \in V'} \mu_{j'} \,.$$

- to meet all demand at $1'$, we need $\lambda_1 + \lambda_2 \geq \mu_{1'}$

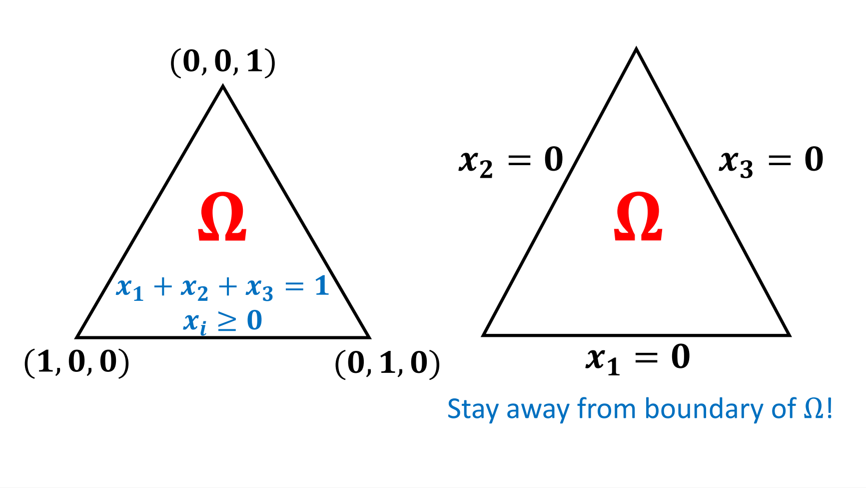
alternate view: walking in the simplex

- State $X(t) = (X_i(t))_{i \in V}$: number of units in each station i
  - state space: $X(t) \in \Omega_K \triangleq \{\mathbf{x} : \mathbf{x} \geq 0, \mathbf{1}^T\mathbf{x} = K\}$
  - normalized state space: $\Omega \triangleq \Omega_1$
  - avoid request drop $\Leftarrow$ avoid hitting boundary



alternate view: walking in the simplex

- State $X(t) = (X_i(t))_{i \in V}$: number of units in each station i
  - state space: $X(t) \in \Omega_K \triangleq \{\mathbf{x} : \mathbf{x} \geq 0, \mathbf{1}^T \mathbf{x} = K\}$
  - normalized state space: $\Omega \triangleq \Omega_1$
  - avoid request drop $\Leftarrow$ avoid hitting boundary



new performance metric: request-drop exponent

$$\gamma(\vec{q}) = -\lim_{K \to \infty} \frac{1}{K} \log \left(\text{fraction of requests dropped under } \pi_K(\vec{q})\right)$$

complete resource pooling (strong Hall's condition)

Assumption

$$\forall J \subset V', \;\; \sum_{i \in \partial(J)} \lambda_i > \sum_{j' \in J} \mu_{j'} \qquad (\star)$$

complete resource pooling (strong Hall's condition)

Assumption

$$\forall J \subset V', \quad \sum_{i \in \partial(J)} \lambda_i > \sum_{j' \in J} \mu_{j'} \qquad (\star)$$

- Hall's theorem: there exists a policy such that **no demand is dropped in fluid limit** $\Leftrightarrow$ **the assumption holds**

complete resource pooling (strong Hall's condition)

Assumption

$$\forall J \subset V', \quad \sum_{i \in \partial(J)} \lambda_i > \sum_{j' \in J} \mu_{j'} \qquad (\star)$$

- Hall's theorem: there exists a policy such that **no demand is dropped in fluid limit** $\Leftrightarrow$ **the assumption holds**

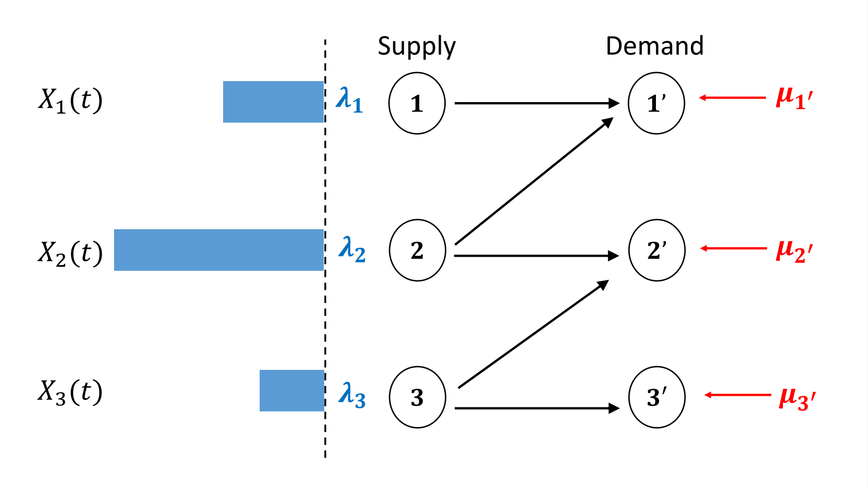
proposition

if $(\star)$ holds with equality, exponential decay in request-drops is impossible

- what if $(\star)$ holds with strict inequality?

the MaxWeight policy

MaxWeight policy: greedily dispatch from $\text{argmax}_{i \in \partial(j')} X_i(t)$

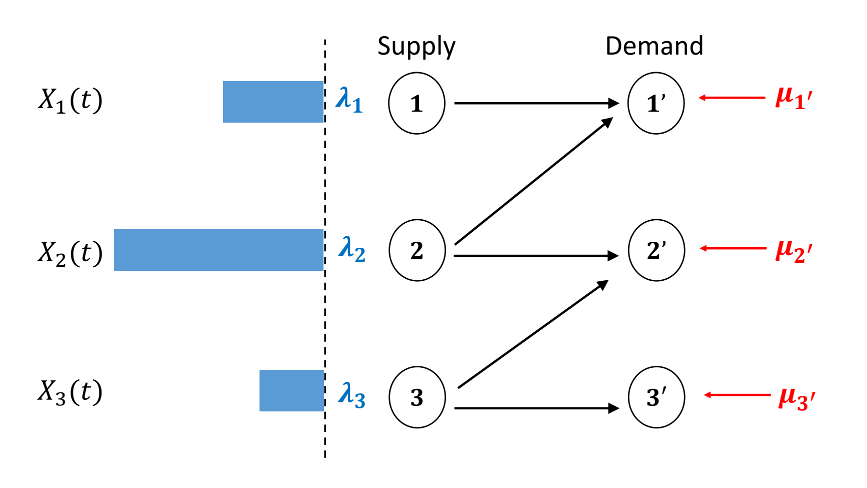


- minimizes delay in open queueing networks in heavy traffic[1]

[1] Dai & Lin 2005, Meyn 2009, Maguluri & Srikant 2016, etc.

the MaxWeight policy

MaxWeight policy: greedily dispatch from $\text{argmax}_{i \in \partial(j')} X_i(t)$



- minimizes delay in open queueing networks in heavy traffic[1]
- sub-optimal in our setting: compared with always dispatch from 1

[1] Dai & Lin 2005, Meyn 2009, Maguluri & Srikant 2016, etc.

scaled MaxWeight: a family of policies

idea: give each location a weight w_i

- $w_i > 0, \sum_{i \in V} w_i = 1$

scaled MaxWeight: a family of policies

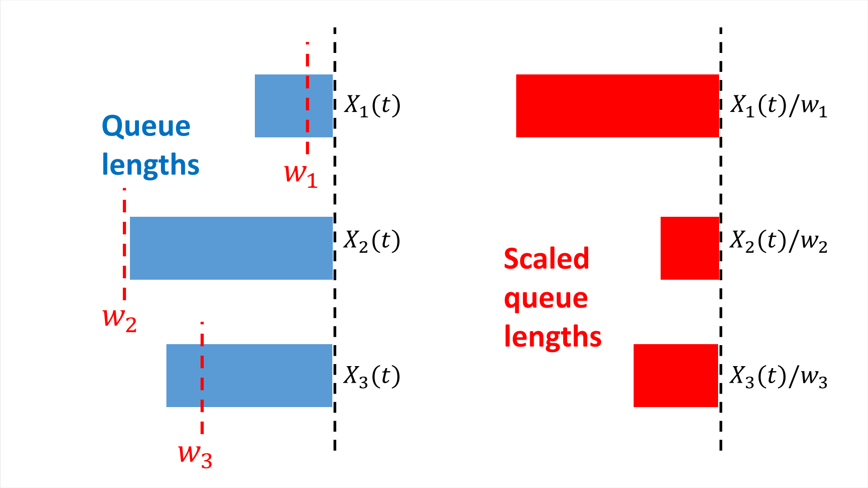
idea: give each location a weight w_i

- $w_i > 0, \sum_{i \in V} w_i = 1$
- **SMW(w)**: for demand at j', dispatch from $\text{argmax}_{i \in \partial(j')} \frac{X_i(t)}{w_i}$

scaled MaxWeight: a family of policies

idea: give each location a weight w_i

- $w_i > 0, \sum_{i \in V} w_i = 1$
- **SMW(w)**: for demand at j', dispatch from $\text{argmax}_{i \in \partial(j')} \frac{X_i(t)}{w_i}$
- previous example: $w_1 \to 0$



how does SMW(**w**) perform?

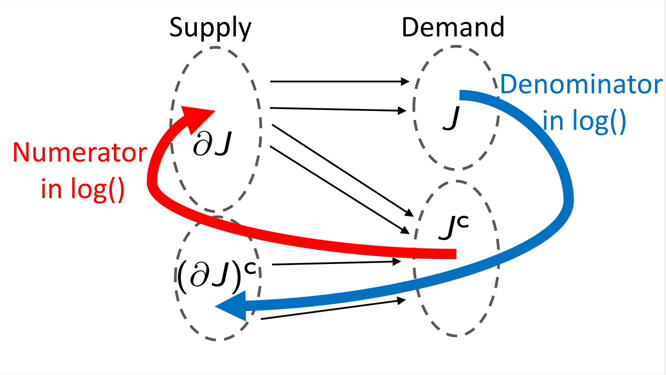
main result: achievability

theorem [B, Kanoria & Qian 2018]

if $(\star)$ is satisfied, then SMW($\mathbf{w}$) for any $\mathbf{w} \in \text{relint}(\Omega)$ results in exponential decay of dropped-requests; in particular

$$\gamma(SMW(\mathbf{w})) = \min_{J \in \mathcal{J}} (\mathbf{1}_{\partial(J)}^T \mathbf{w}) \log \left(\frac{\sum_{j' \notin J} \sum_{k \in \partial(J)} \phi_{j'k}}{\sum_{j' \in J} \sum_{k \notin \partial(J)} \phi_{j'k}} \right)$$

where $\mathcal{J} \triangleq \left\{ J \subsetneq V' : \sum_{j' \in J} \sum_{k \notin \partial(J)} \phi_{j'k} > 0 \right\}$



main result: converse

theorem [B, Kanoria & Qian 2018]

for any instance satisfying ($\star$), let $\gamma^* \triangleq \sup_{\mathbf{w}:\mathbf{w}\in \text{relint}(\Omega)} \gamma(\mathbf{w})$
then any state dependent policy satisfies

$$-\liminf_{K\to\infty} \frac{1}{K} \log\left(\text{fraction of dropped requests with } K \text{ supplies}\right) \leq \gamma^*,$$

corollary

there is an SMW policy that achieves a demand drop exponent as close as desired to the optimal one.

implications of our results

- scaled MaxWeight: one policy to rule them all!

implications of our results

- scaled MaxWeight: one policy to rule them all!
- can adapt to available system information
  - if we know nothing, use vanilla MaxWeight; resulting exponent is within factor n of optimal

implications of our results

- scaled MaxWeight: one policy to rule them all!
- can adapt to available system information
  - if we know nothing, use vanilla MaxWeight; resulting exponent is within factor n of optimal
  - if we know ϕ perfectly, solve for optimal $\mathbf{w}^*$

implications of our results

- scaled MaxWeight: one policy to rule them all!
- can adapt to available system information
  - if we know nothing, use vanilla MaxWeight; resulting exponent is within factor n of optimal
  - if we know ϕ perfectly, solve for optimal $\mathbf{w}^*$
  - if we have partial information . . .

implications of our results

- scaled MaxWeight: one policy to rule them all!
- can adapt to available system information
  - if we know nothing, use vanilla MaxWeight; resulting exponent is within factor n of optimal
  - if we know ϕ perfectly, solve for optimal $\mathbf{w}^*$
  - if we have partial information ...
- separation of pricing and dispatch: slowly changing pricing + real-time dispatch sufficient in many cases

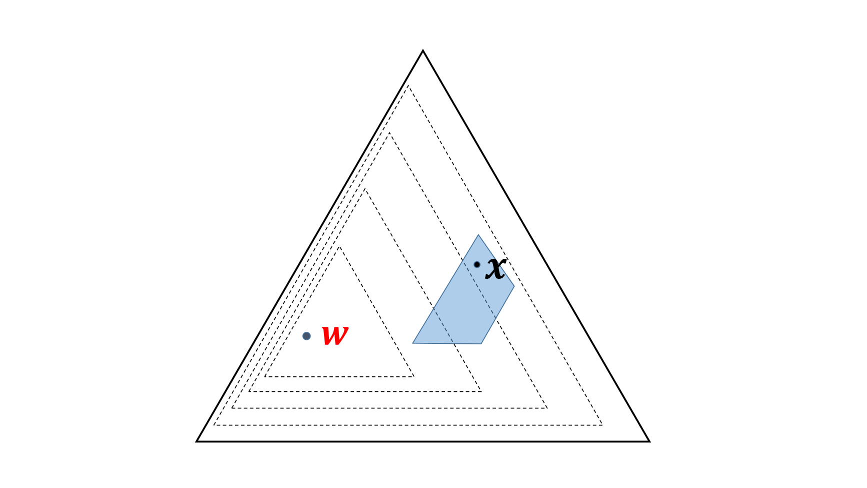
brief proof outline

technical difficulty

- compared to heavy traffic: n-**dimensional** problem
- compared to open networks: **no state-space ordering**

we use a large-deviations analysis with policy-dependent Lyapunov fn

proof highlight: custom Lyapunov function



- SMW pushes the state towards $\mathbf{w}$
- construct a Lyapunov function $L_{\mathbf{w}}(x) \geq 0$ such that $L_{\mathbf{w}}(\mathbf{w}) = 0$, and $L_{\mathbf{w}} = 1$ on $\partial\Omega$, and L is scale invariant about $\mathbf{w}$
- SMW($\mathbf{w}$) performs steepest descent w.r.t. $L_{\mathbf{w}}$

$$L_{\mathbf{w}}(X) = 1 - \min_{i \in V} \frac{X_i}{w_i}$$

summary

- stochastic models for dispatch: interesting insights + cool new theory!
- near-optimal state-independent dispatch via flow relaxations
- state-dependent SMW policies lead to exponentially decreasing fraction of dropped requests
  - can adapt to available information
  - pricing-dispatch separation principles

open questions

- state-dependent dispatch with travel-times
- elementary proofs of exponential decay?
- adapting policy to changing information
- dispatch for pooling