Models, Performance and Optimization of Ride-Hailing Platforms

Costas Courcoubetis (SUTD) and Antonis Dimakis (AUEB)

Outline

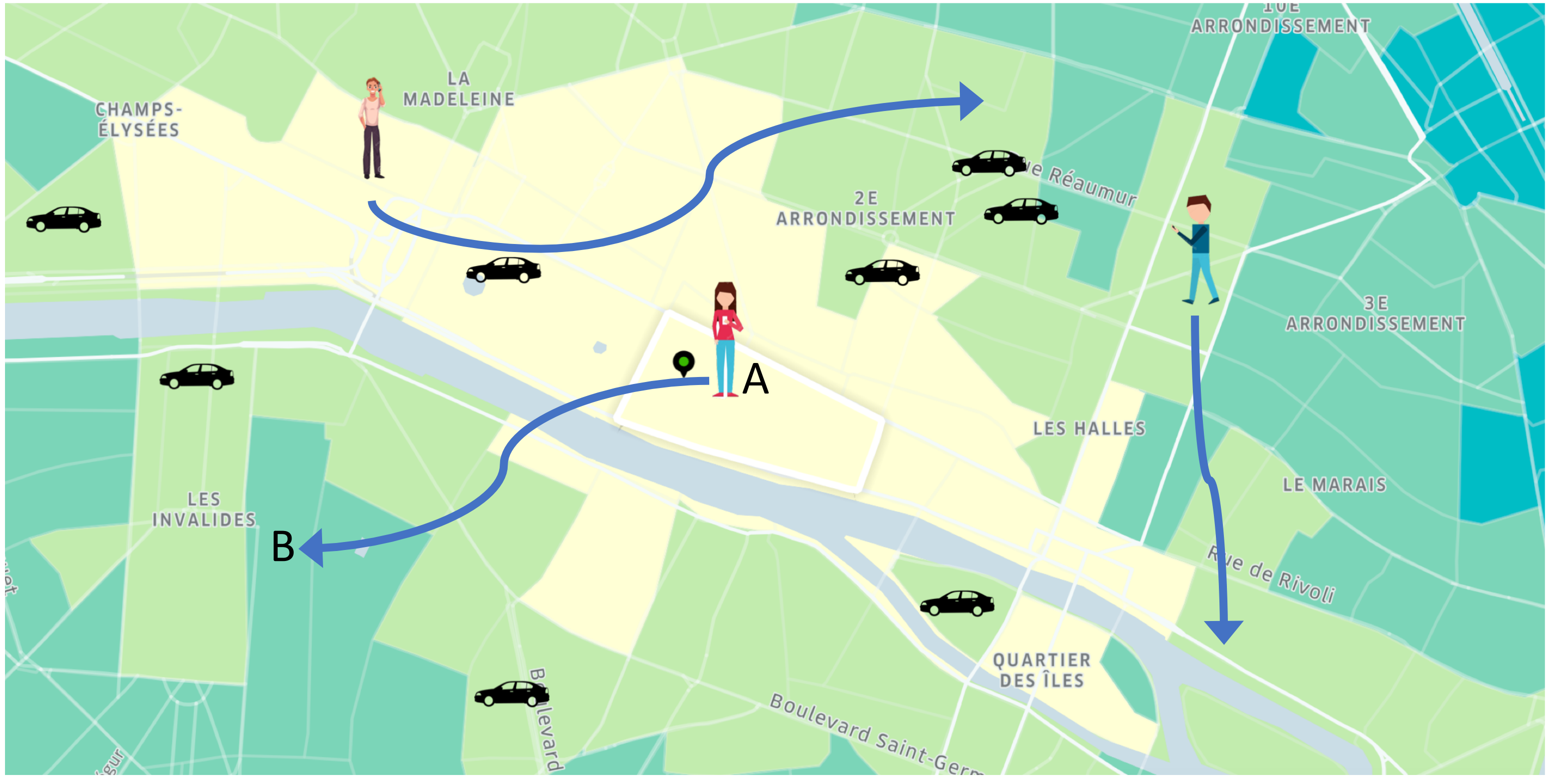
- Ridesharing
- A model of passenger queueing
- Stability region
- Stability with static prices
- A surge-type of dynamic pricing
- Throughput optimality of surge pricing
- Simulations

Ride-hailing

- Original idea: offer transportation services by exploiting *existing* mobility through information technologies
- Ended up being mostly used by professional car drivers which pursue profit maximization
- Has greatly disrupted transportation industry all around the world
- Generic model:
  - Passengers and car drivers subscribe to the platform
  - Platform connects them when transportation is requested

Ride-hailing: the picture

1. User from A opens app
2. Nearest drivers and pickup times reported
3. User queries ride to B
4. Ride fee reported
5. User request ride
6. A driver is matched to the user
7. Driver picks up user
8. User transported to B



Algorithms & objectives

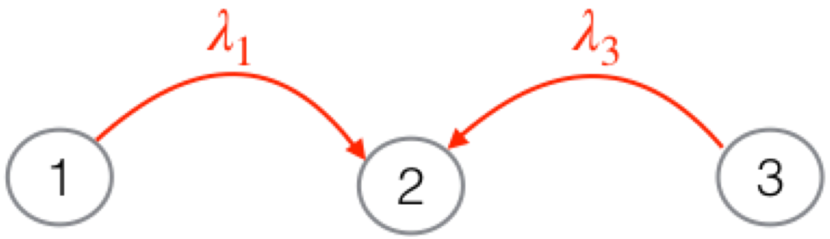
- Overall system operation determined by the complex interplay of multiple players
- Passengers' objective: *find a cheap ride, fast*
  - How long to wait? Maximum acceptable ride fee? When to request a ride?
- Drivers' objective: *maximize profits*
  - Enter platform? If yes, when & where?
  - Accept ride or not?
  - Where to "hunt" for rides?
- Platform's objectives: *maximize profits, throughput*
  - **Driver-user matching**
  - **Driver compensation & passenger fee**
    - Should pickup location matter?
    - Dynamic or static?
- Devise algorithms which lead to "good" operating points

Relation to existing work

- *Aggregate* throughput or platform profit maximization
  - No queueing for passengers
  - Not all potential passenger demand is served
  - No driver incentives considered:
    - [Özkan&Ward'16]: driver-customer matching, asymp. opt. LP-based matching
    - [Braverman et al.'16]: free driver mobility, asymp. opt. routing
  - Analysis of equilibrium between platform, drivers & customers
    - [Riquelme et al.'16]: drivers decide entry – not routing, static prices as good as dynamic,
    - [Bimpikis et al.'16]: drivers decide routing, effect of balanced demand on platform profits.
- We consider throughput *in each region* over *shorter* timescales
  - timescale=number of drivers in the system is constant
  - ➢ *What throughput combinations are possible and how to create incentives for their selection?*
  - ➢ *Stability: is a given passenger demand serviceable?*
  - [Bimpikis et al.'16] tackle a similar problem but under special symmetry assumptions
- Queueing model of passengers waiting for pick-up
  - ➢ *Interpret surge-pricing as stability achieving dynamic policy*

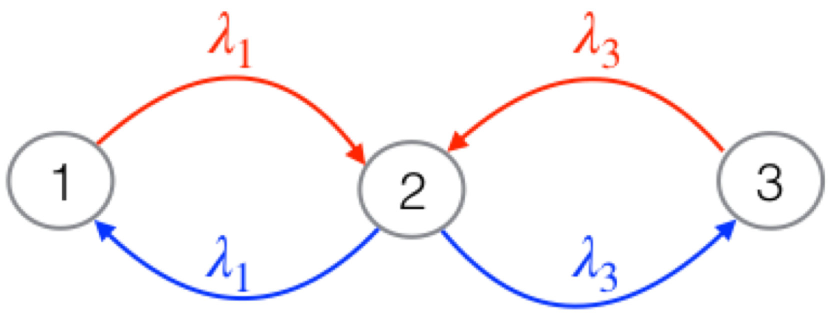
Toy example

- What is the minimum number of drivers d to serve passenger demand rates λ_1, λ_3?



Toy example

- What is the minimum number of drivers d to serve passenger demand rates λ_1, λ_3?

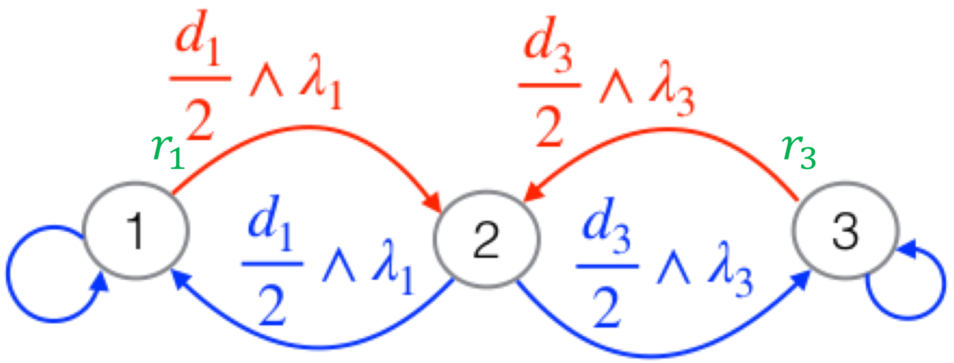


> ➢ *Stability condition:*
> $2\lambda_1 + 2\lambda_3 \leq d$

- How to enforce optimal mobility pattern for free drivers at 2?

Toy example

- What is the minimum number of drivers d to serve passenger demand rates λ_1, λ_3?



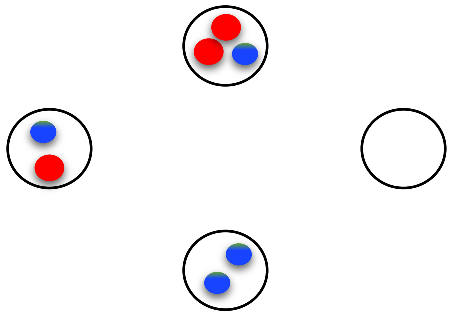
> ➢ *Stability condition:*
> $2\lambda_1 + 2\lambda_3 \leq d$

- How to enforce optimal mobility pattern for free drivers at 2?
  - Platform gives reward r_i to drivers giving rides from i
  - $\frac{r_1\lambda_1}{d_1 \vee 2\lambda_1} = \frac{r_3\lambda_3}{d_3 \vee 2\lambda_3}, d_1 + d_3 = d$ imply a **unique** mobility pattern (d_1, d_3) as driver response

> ➢ *Choose rewards which give* $\frac{d_i}{2} > \lambda_i$, *e.g.,* $r_1 = r_3$

- Can rewards be selected dynamically/distributedly to stabilize system?
- If drivers are not rational optimizers?

Driver model



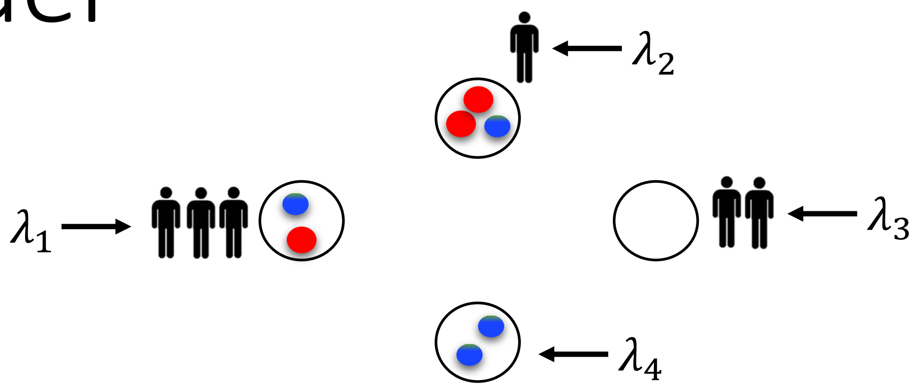
$d = 7, r = 4$

= busy driver

= free driver

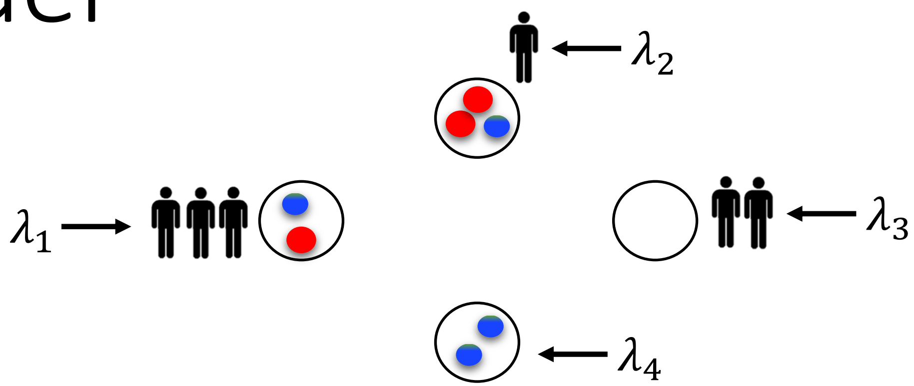
- d drivers
- r regions
- A continuous time controlled semi-Markov process models driver mobility
- Each driver entering region i:
  1. Becomes *busy* with probability $\Theta_i(t)$, or stays *free*
  2. Decide where to transit next:
     - If *busy*, choose j with probability q_{ij} (given by passenger's destination)
     - If *free*, choose j with probability $P_{ij}(t)$ (decided by driver)
  3. Actual transition happens after a random exponential time with mean $1/\mu_{ij}$
- *Policy process* $(P_{ij}(t), \Theta_i(t), i, j = 1, \ldots, r, t \geq 0)$.

Queueing model



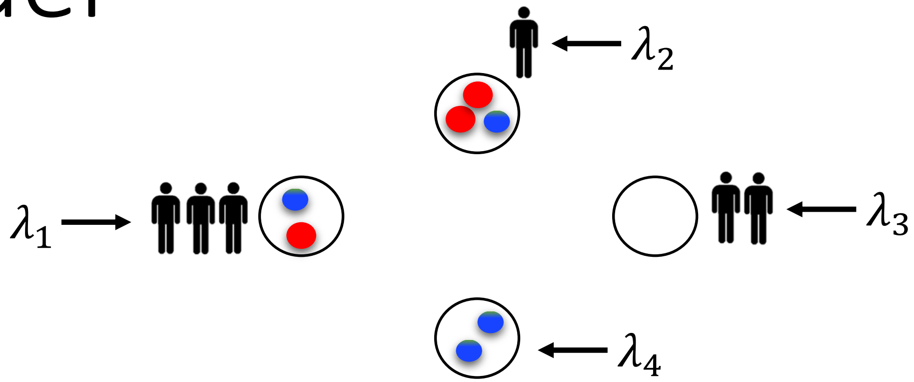
- Passengers arrive to region i according to a Poisson process with rate λ_i
  - q_{ij} proportion has destination region j
- $Q_i(t)$ = length of customer queue in region i
- An arriving driver which becomes busy, decreases $Q_i(t)$ (if nonzero) by one
- Drivers may be allowed to become busy when $Q_i(t) = 0$: ***virtual service***
  - Driver follows transition according to (q_{ij}), queue length remains 0
- Policy process can depend arbitrarily on past histories (e.g., queue sizes) up to $t-$.
- Π: set of all policy processes
- Π_s: set of *static* policy processes $(P_{ij}(t) = p_{ij}, \Theta_i(t) = \theta_i$ for some $(p_{ij}), (\theta_i))$
- Π_0: set of policy processes *not allowing virtual service* $(Q_i(t-) = 0 \Rightarrow \Theta_i(t) = 0)$
- Π_{0w}: *work conserving policy* process not allowing virtual service (also $Q_i(t-) \neq 0 \Rightarrow \Theta_i(t) = 1$)

Queueing model



- *Stability:* $\limsup_t \frac{E\left[Q_i(t)\right]}{t} = 0\ , \forall i$ *for some initial distribution of drivers' locations*
  - passenger departure rate = arrival rate

Queueing model



- *What is the largest set of arrival rates* $\boldsymbol{\lambda} = (\lambda_i, i = 1, \ldots, r)$ *that can be supported without violating stability?*
  - Λ*: stability region for all policies (*Π*)*
  - Λ_s*: stability region for static policies (*Π_s*)*
  - Λ_0*: stability region for policies without virtual service (*Π_0*)*
  - $\Lambda_{0\mathrm{w}}$*: stability region for work conserving policy without virtual service (*$\Pi_{0\mathrm{w}}$*)*

- *How can the platform influence drivers to pick optimal policies?*

Static mobility pattern

Free driver routing given by (p_{ij})

Driver balance equations

Let

- b_i: rate of drivers becoming busy in region i
- f_i: rate of drivers idling in region i

for any policy which the limits exist.

- Flow of drivers entering and leaving each region must balance:

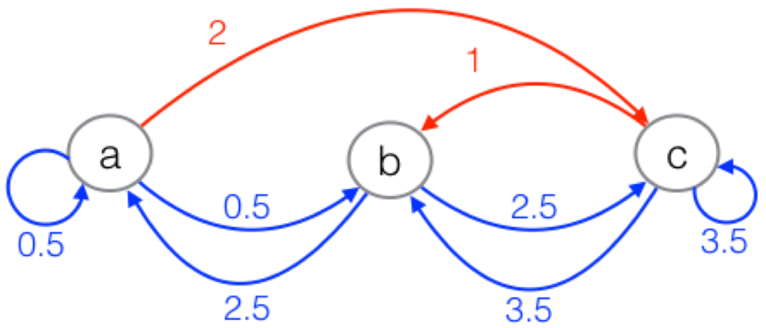
$$f_i + b_i = \sum_j f_j p_{ji} + \sum_j b_j q_{ji} \ , \forall i$$

- Total number of drivers is d:

$$\sum_{i,j} \frac{b_i q_{ij}}{\mu_{ij}} + \sum_{i,j} \frac{f_i p_{ij}}{\mu_{ij}} = d$$

Driver Balance Equations

- Example: 3 regions, $d = 18$ drivers
  - Mobility pattern: symmetric random walk
  - Given busy rates $b_a = 2, b_c = 1$
  - Uniquely implied f_i's



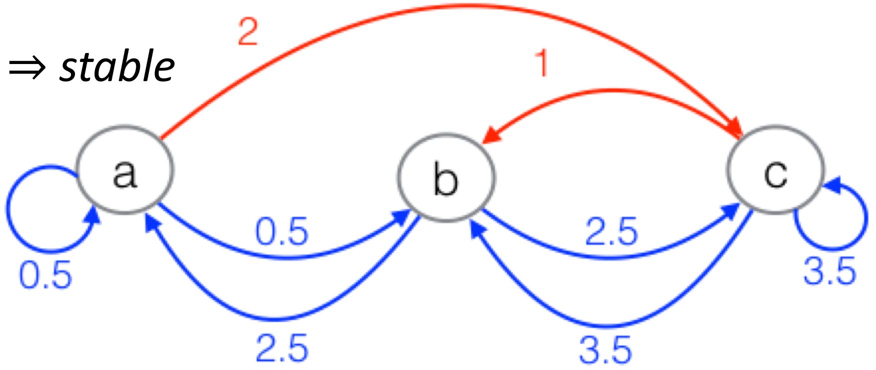
➢ *There exists a policy which implements given b_i's: $\theta_i = \frac{b_i}{b_i + f_i}$*

Stability for a given mobility pattern

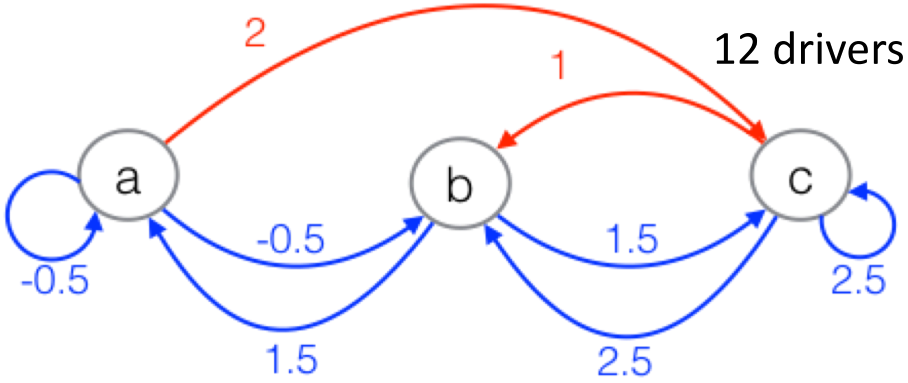
- Given arrival rates $\boldsymbol{\lambda}$ and a mobility pattern does there exist a stable policy?
- Suffices to find a nonnegative solution $(\boldsymbol{b}, \boldsymbol{f}) = (b_i, f_i, i = 1, \dots, r)$ to the driver balance equations such that $\boldsymbol{\lambda} \le \boldsymbol{b}$, for a stable static policy to exist for some **initial driver placement**
  - If no virtual service allowed, must have $\boldsymbol{\lambda} = \boldsymbol{b}$
- Conversely, if no nonnegative solution exists *no* stable policy in Π exists

- Example: $\lambda_a = 2, \lambda_c = 1, q_{ac} = 1, q_{cb} = 1$, without virtual service

18 drivers ⇒ *stable*



12 drivers ⇒ *unstable*



Stability for a given mobility pattern

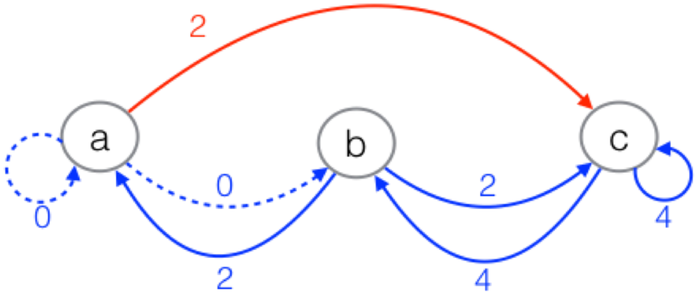
Theorem 1:

i. $\Lambda = \Lambda_s$ *is the set of arrival rates* $\boldsymbol{\lambda}$ *such that* $(\color{red}{\boldsymbol{b}}, \color{blue}{\boldsymbol{f}})$ *exist with* $\boldsymbol{\lambda} \leq \color{red}{\boldsymbol{b}}, \color{blue}{\boldsymbol{f}} \geq 0$, *and the driver balance equations are satisfied, i.e.,*

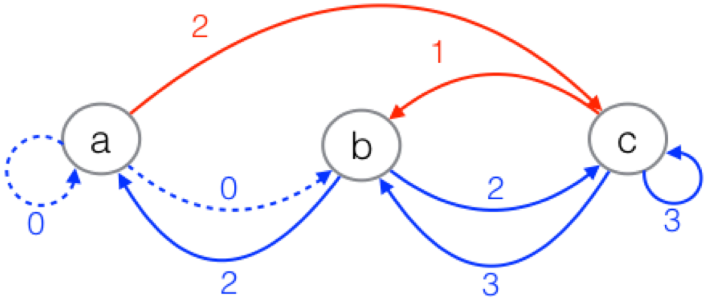
$$f_i + b_i = \sum_j f_j p_{ji} + \sum_j b_j q_{ji} \,, \forall i \qquad \sum_{i,j} \frac{b_i q_{ij}}{\mu_{ij}} + \sum_{i,j} \frac{f_i p_{ij}}{\nu_{ij}} = d$$

ii. $\Lambda_0 = \Lambda_{0w}$ *is the set of arrival rates as above, with* $\boldsymbol{\lambda} \leq \color{red}{\boldsymbol{b}}$ replaced by $\boldsymbol{\lambda} = \color{red}{\boldsymbol{b}}$

- An increase in demand could lower the necessary # of drivers (i.e., $\Lambda_0 \subsetneq \Lambda$)



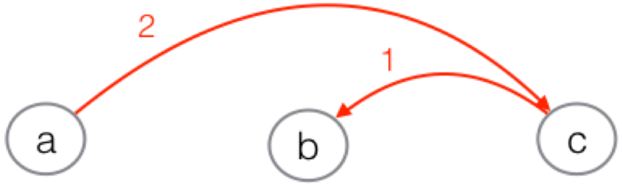
Min # of drivers= 16 $>$



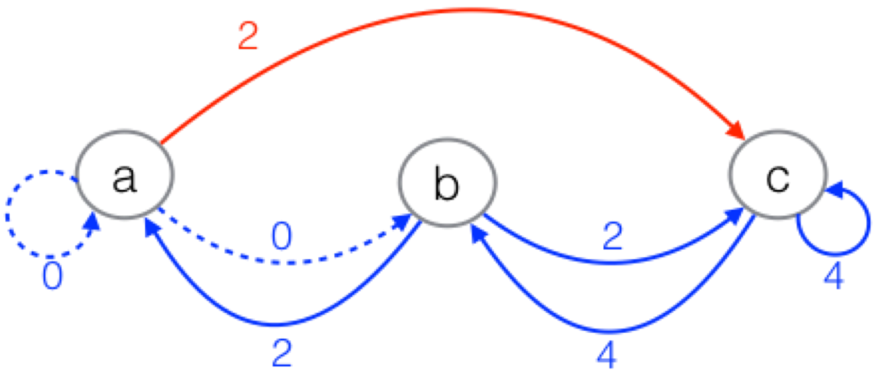
Min # of drivers= 15

Stability condition for Λ_0, Λ_{0w}

- Not clear how system characteristics affect stability in driver balance equations
- Simpler necessary and sufficient condition for stability in Π_0, Π_{0w}
- Example:
  - What is the minimum number of drivers d for stability?
  - $\lambda_a = 2, \lambda_c = 1$



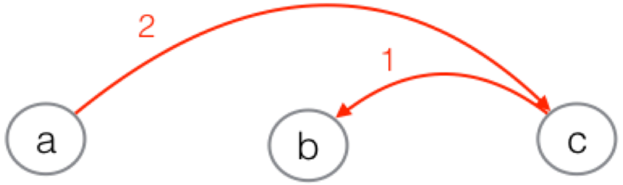
  - Determine *minimum # of drivers such that* <u>each</u> *origin-destination pair is stable without virtual service*:



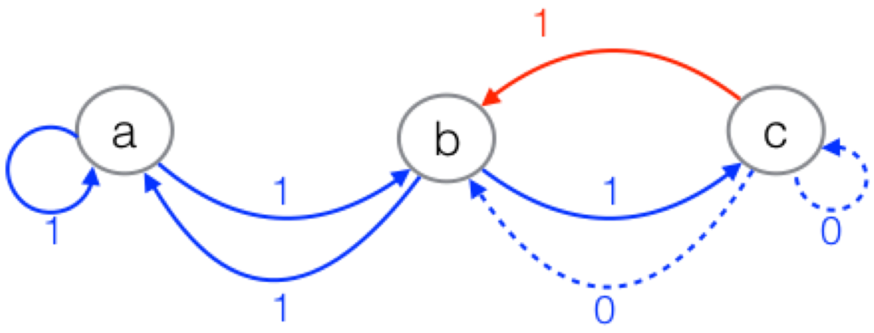
Min # of drivers $= \frac{\lambda_i q_{ij}}{\mu_{ij}} + \lambda_i q_{ij} T_{ji} = 16$

Stability condition for Λ_0, Λ_{0w}

- Not clear how system characteristics affect stability in driver balance equations
- Simpler necessary and sufficient condition for stability in Π_0, Π_{0w}
- Example:
  - What is the minimum number of drivers d for stability?
  - $\lambda_a = 2, \lambda_c = 1$



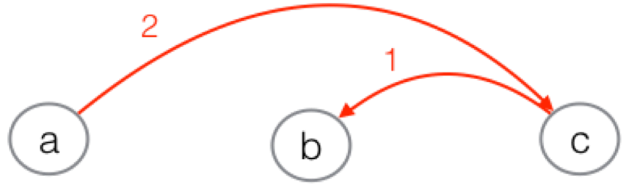
  - Determine *minimum # of drivers such that* <u>each</u> *origin-destination pair is stable without virtual service*:



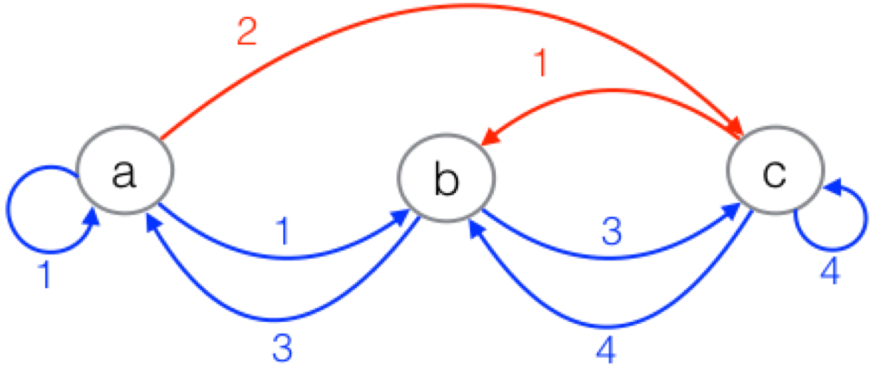
Min # of drivers $= \frac{\lambda_i q_{ij}}{\mu_{ij}} + \lambda_i q_{ij} T_{ji} = 5$

Stability condition for Λ_0, Λ_{0w}

- Not clear how system characteristics affect stability in driver balance equations
- Simpler necessary and sufficient condition for stability in Π_0, Π_{0w}
- Example:
  - What is the minimum number of drivers d for stability?
  - $\lambda_a = 2, \lambda_c = 1$



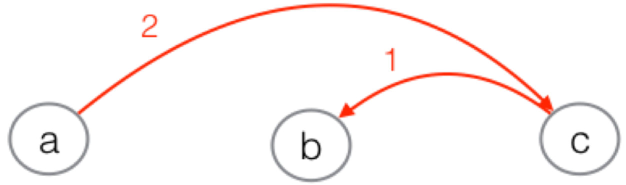
  - Determine *minimum # of drivers such that* each *origin-destination pair is stable without virtual service*:



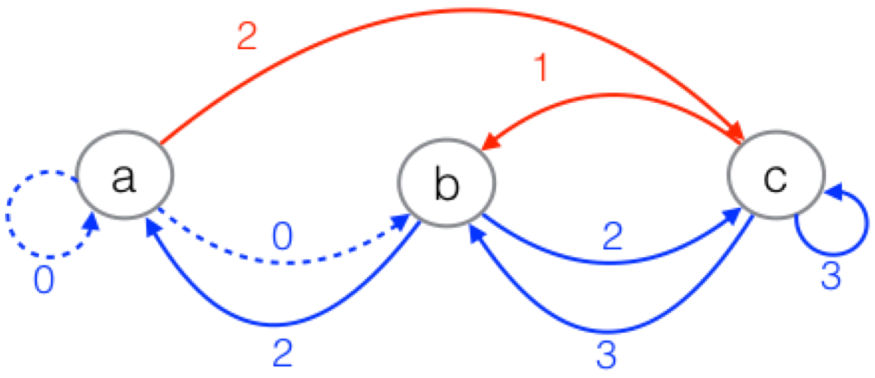
Min # of drivers $= \sum_{ij} \frac{\lambda_i q_{ij}}{\mu_{ij}} + \sum_{ij} \lambda_i q_{ij} T_{ji} = 21$

Stability condition for Λ_0, Λ_{0w}

- Not clear how system characteristics affect stability in driver balance equations
- Simpler necessary and sufficient condition for stability in Π_0, Π_{0w}
- Example:
  - What is the minimum number of drivers d for stability?
  - $\lambda_a = 2, \lambda_c = 1$



  - Determine *minimum # of drivers such that* each *origin-destination pair is stable without virtual service*:



$$\text{Min \# of drivers} = \sum_{ij} \frac{\lambda_i q_{ij}}{\mu_{ij}} + \sum_{ij} \lambda_i q_{ij} T_{ji} - overlap\ term = 15$$

➢ *Stability condition*: $15 \leq d$

Stability condition for Λ_0, Λ_{0w}

- d = # of drivers
- (p_{ij}): mobility pattern having *a **single recurrent class***
- $\boldsymbol{\lambda} = (\lambda_i)$*: arrival rates satisfying:* $\lambda_i q_{ij} > 0$*,* j *communicates with* k *(wrt.* p*)* $\Rightarrow$ k *communicates with* i *(wrt.* p*) (*)*
- T_{ji} = mean travel time from j to i if always free
- $1/\nu_k$= mean time to leave k when free
- π_k = portion of time spent in k if always free
- f_k^{ij} = rate of free drivers out of k under a unit busy flow of drivers from i to j

Theorem 2: $\boldsymbol{\lambda} \in \Lambda_0 = \Lambda_{ow}$ *if and only if*

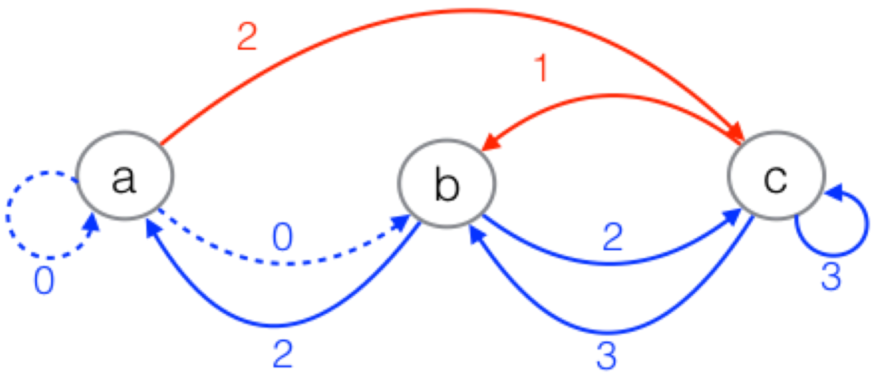
$$\underbrace{\sum_{i,j} \frac{\lambda_i q_{ij}}{\mu_{ij}}}_{\text{necessarily busy}} + \underbrace{\sum_{i,j} \lambda_i q_{ij} T_{ji}}_{\text{upper bound for free}} - \underbrace{\min_{k:\pi_k \neq 0} \frac{\sum_{i,j} \lambda_i q_{ij} f_k^{ij}}{\pi_k \nu_k}}_{\text{overlap term}} \leq d$$

Optimizing over p_{ij}'s

Incentivizing driver mobility

Mobility pattern optimization

- Can carry more passengers by optimizing the routing of free cars



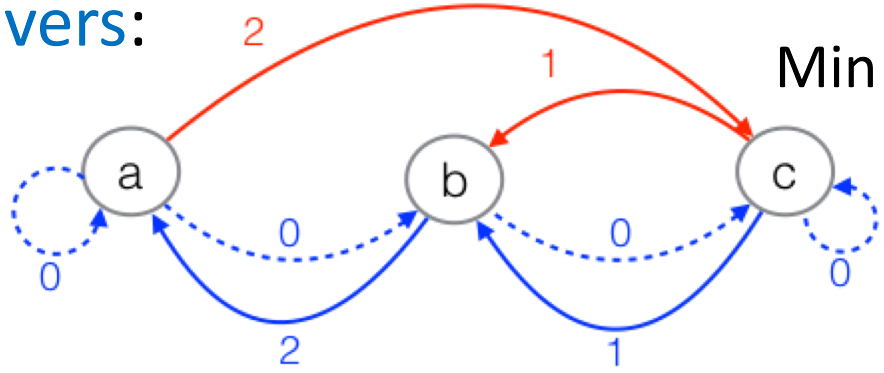
Min # of drivers= 15

- Optimal transshipment of free drivers:

$$\min \sum_{i,j} \frac{f_{ij}}{\mu_{ij}}$$

$$\text{s.t.} \;\; \lambda_i + \sum_j f_{ij} = \sum_j \lambda_j q_{ji} + \sum_j f_{ji} \,, \forall i$$

$$\text{over} \;\; f_{ij} \geq 0, i, j = 1, \ldots, r$$



Min # of drivers= 8

Stability region with flexible mobility

- b_i: rate of drivers becoming busy in region i (incl. virtual service)
- f_{ij}: rate of free drivers flowing from i to j
- Any policy satisfies the driver balance equations:

$$\sum_j f_{ij} + b_i = \sum_j f_{ji} + \sum_j b_j q_{ji} \; \forall i$$

$$\sum_{i,j} \frac{b_i q_{ij}}{\mu_{ij}} + \sum_{i,j} \frac{f_{ij}}{\mu_{ij}} = d$$

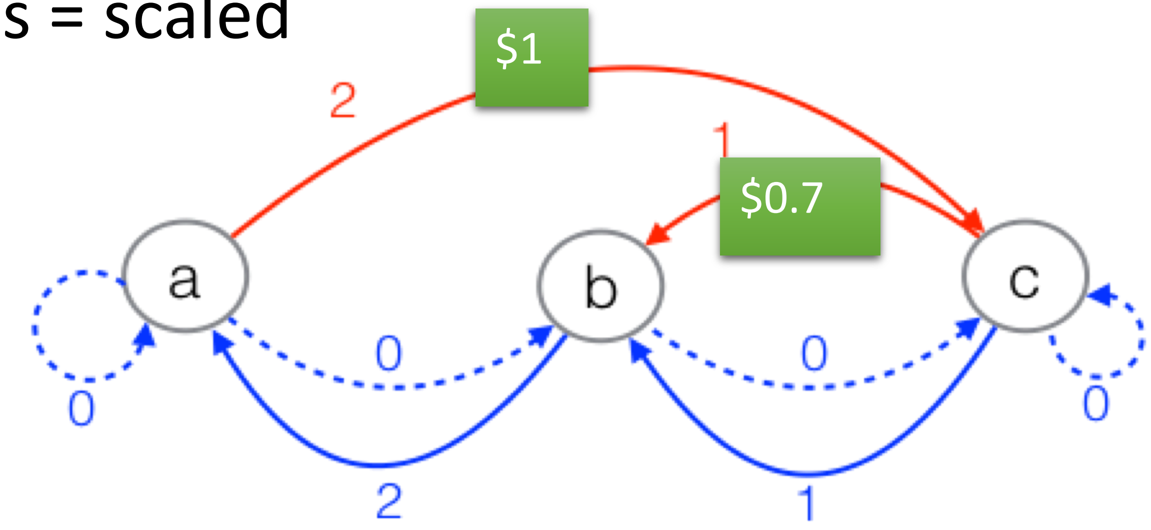
Theorem 3: *$\Lambda = \Lambda_s = \Lambda_0 = \Lambda_{0w}$ is the set of arrival rates $\boldsymbol{\lambda}$ such that $(\boldsymbol{b}, \boldsymbol{f})$ exist with $\boldsymbol{\lambda} = \boldsymbol{b}, \boldsymbol{f} \geq \mathbf{0}$, and the driver balance equations are satisfied*

Proof: $\Lambda_0 = \Lambda_{0w} \subseteq \Lambda$ *by Theorem 2*

Any stable process satisfies driver balance eq with $\boldsymbol{b} = \boldsymbol{\lambda}$

Optimizing mobility through rewards

- How to influence drivers to follow (throughput) optimal mobility patterns?
- Drivers maximize time average reward rates
  - Offer rewards to drivers depending on the region a passenger is picked up from
  - Reward values = scaled



Reward-optimizing policies

- A passenger pick-up from i generates reward at rate r_i for the duration of transport
- Simplification: *busy decision depends on current location but not on queue state*
  - A busy driver who ends up not serving gets no reward and moves freely
- Let ϕ_i = probability a busy driver at i *actually serves* a passenger
- Drivers choose busy rates $\boldsymbol{b} = (b_i)$ which maximize *average reward rate*:

$$
\begin{aligned}
\mathrm{DRIVER}(r,\phi) : \max \sum_i \frac{r_i}{\mu_i}\phi_i b_i \\
\text{s.t. } \phi_i b_i + \sum_j f_{ij} = \sum_j \phi_j b_j q_{ji} + \sum_j f_{ji}\,, \\
\sum_i \frac{\phi_i b_i}{\mu_i} + \sum_{i,j} \frac{f_{ij}}{\mu_{ij}} = d\,, \\
\sum_j f_{ij} \geq (1-\phi_i) b_i\,, \forall i \in R\,, \\
\text{over } b_i, f_{ij} \geq 0, i, j, \in R\,.
\end{aligned}
$$

Reward-optimizing policies

Theorem 4: *If $\boldsymbol{\lambda} \in \Lambda$ there exist reward rates $\boldsymbol{r} = (r_i)$, service probabilities $\boldsymbol{\phi} = (\phi_i)$, and busy rates $\boldsymbol{b}$ such that:*

1. *$\boldsymbol{b}$ maximizes DRIVER($\boldsymbol{r}, \boldsymbol{\phi}$), and*
2. *$\lambda_i = b_i \phi_i$ for all i*

- *Proof*: use dual of

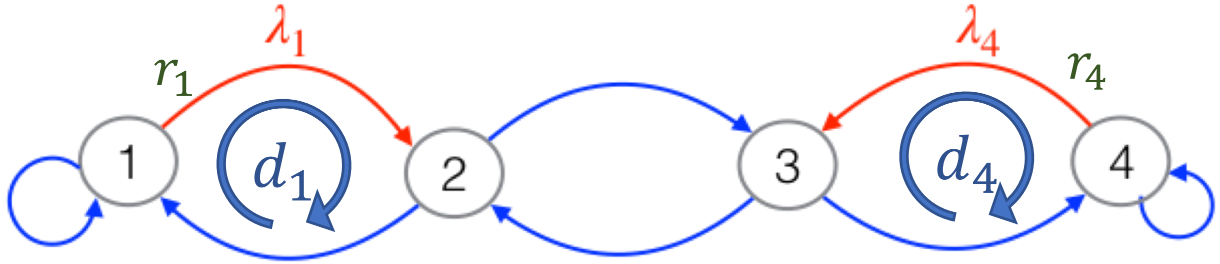
$$\begin{aligned} \max \quad & \rho \\ \text{s.t.} \quad & \rho \boldsymbol{\lambda} \leq z \boldsymbol{b} \\ & (\boldsymbol{b}, \boldsymbol{f}) \text{ feasible in DRIVER}(\mathbf{0}, z\mathbf{1}) \\ \text{over} \quad & \rho, \boldsymbol{b}, \boldsymbol{f} \geq 0. \end{aligned}$$

  for appropriate $z \in [0,1]$

- Caveats:
  - Rewards depend on system parameters (e.g., demand) which need to be estimated
  - It is important for $\boldsymbol{r}$ to fully determine $\boldsymbol{\phi}$, $\boldsymbol{b}$

Uniqueness of service rates

- *Q: Do rewards determine throughput uniquely?*
- Example: $r_1 = r_4$



➢ If $d_1 < 2\lambda_1, d_4 < 2\lambda_4$ then *multiple* d_1, d_4 possible

- Reward per driver = $r_1/2 = r_4/2$

➢ If stability condition $2\lambda_1 + 2\lambda_4 < d_1 + d_4$ satisfied then d_1, d_4 unique

- $\frac{r_1\lambda_1}{d_1 \vee 2\lambda_1} = \frac{r_4\lambda_4}{d_4 \vee 2\lambda_4}, d_1 + d_4 = d$ imply *unique* d_1, d_4

Reward-optimizing policies

***Proposition**: Given $\boldsymbol{\lambda}, \boldsymbol{r}$, the equilibrium $(\boldsymbol{b}, \boldsymbol{\phi})$ in Theorem 4 is unique (modulo regions with zero reward) if feasible driver circulations are composed by independent cycles*

- *Proof*:
  - Equilibrium $(\boldsymbol{b}, \boldsymbol{\phi})$ construction:
    1. Start with zero circulating drivers in every cycle c
    2. Let $D_c =$ mean cycle time of cycle c
    3. Place an infinitesimal amount of drivers on cycle of highest reward per driver $(\max r_c/D_c)$ Continue to 2 until all d drivers are placed
    4. $\boldsymbol{\phi}, \boldsymbol{b}$ completely determined by D_c and $\lambda_i = b_i \phi_i$ for all i
  - Construction generates unique outcome

***Conjecture**: uniqueness also holds in arbitrary systems*

Selection of rewards

***Remark:** The rewards $\boldsymbol{r}$ which incentivize a stable policy, as selected in the last theorem, solve*

$$\boxed{\begin{aligned} \min \quad & \mathrm{DRIVER}(r, \phi*) \\ \text{s.t.} \quad & \sum_i \frac{r_i}{\mu_i}\lambda_i = 1 \\ \text{over } & r_i \geq 0, \forall i \end{aligned}}$$

Charge r_i to customers in location i
Reward by r_i busy driver in location i
Normalized platform revenue = fixed

for some service probabilities $\boldsymbol{\phi}^*$

- DRIVER$(\boldsymbol{r}, \boldsymbol{\phi}^*)$ = maximum reward earned by a reward-optimizing driver under rewards r and service probabilities $\boldsymbol{\phi}^*$
- For a fixed normalized platform revenue, choose the r_i's to minimize average driver profit

Dynamic rewards

- Static rewards require estimates of system characteristics and reward virtual service
- *MaxWeight dynamic policy*: r_i proportional to $Q_i(t)$ at all times t
  - $Q_i(t) =$ # of customers waiting in region i [Tassiulas & Ephremides'92]
- A type of Uber-like surge pricing
- Each driver acts as if she *would* have maximized (time-)average rewards if $\boldsymbol{Q}(\boldsymbol{t})$ did not change
  - Drivers solve DRIVER$(\boldsymbol{Q}(\boldsymbol{t}-), \mathbf{1})$
  - Valid if $Q(t)$ changes on a slower timescale than travel times, e.g., in heavy loads
- Virtual service not rewarded
- Platform needs only to inform drivers about current rewards – no need to keep estimates

Throughput optimality of MaxWeight

Theorem 5: *The MaxWeight policy is stable for all arrival rates in the stability region and for any initial driver placement*

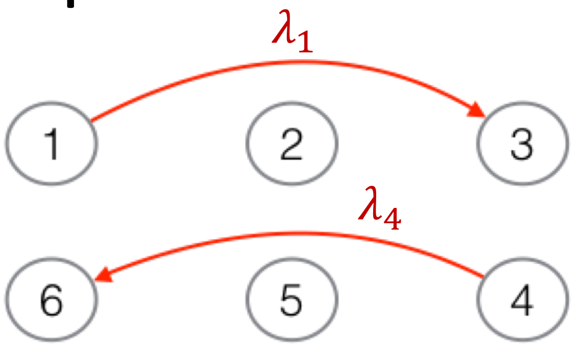
- Proof: $\sum_i \frac{q_i(t)^2}{\mu_i}$ is a Lyapunov function for any fluid limit trajectory

since
$$\frac{d}{dt}\sum_i \frac{q_i(t)^2}{\mu_i} = 2\left[\sum_i \frac{q_i(t)}{\mu_i}\lambda_i - \sum_i \frac{q_i(t)}{\mu_i} b_i(t)\right]$$
$$\leq 2\left[\sum_i \frac{q_i(t)}{\mu_i} b_i - \underbrace{\sum_i \frac{q_i(t)}{\mu_i} b_i(t)}_{\text{Maximum average reward } \mathrm{DRIVER}(\boldsymbol{q}(\boldsymbol{t}), \mathbf{1})}\right] \leq 0$$

($\boldsymbol{\lambda} \in \Lambda \Rightarrow \boldsymbol{\lambda} \leq \boldsymbol{b}$ for some busy rates $\boldsymbol{b}$)

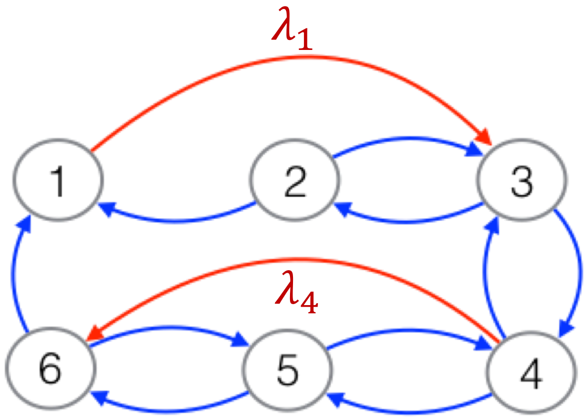
Numerical example

- 6 regions, d drivers
- Assume $\lambda_1 > \lambda_4$



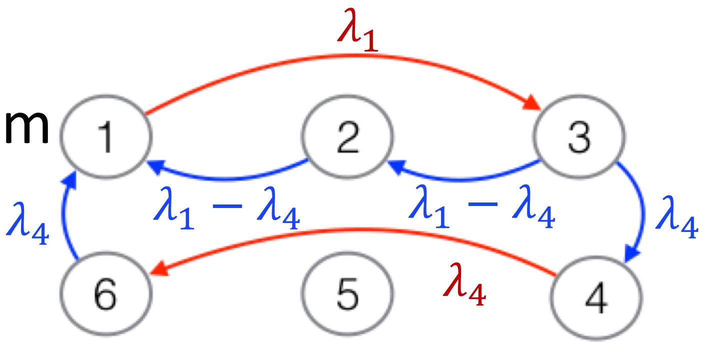
- Free drivers follow a symmetric random walk:

$$10(\lambda_1 + \lambda_4) - 12\min(\lambda_1, \lambda_4) \leq d$$



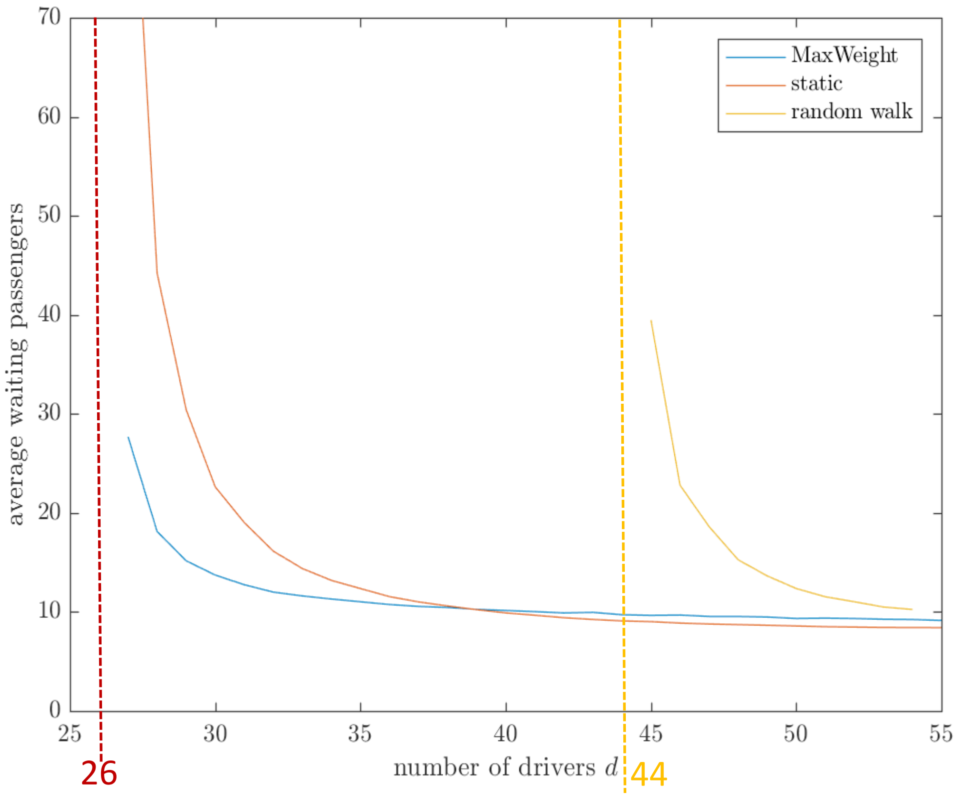
- Throughput optimal Π_{0w} policy:
  - Routing obtained by solving the transshipment problem

$$2(\lambda_1 + \lambda_4) + 2\max(\lambda_1, \lambda_4) \leq d$$

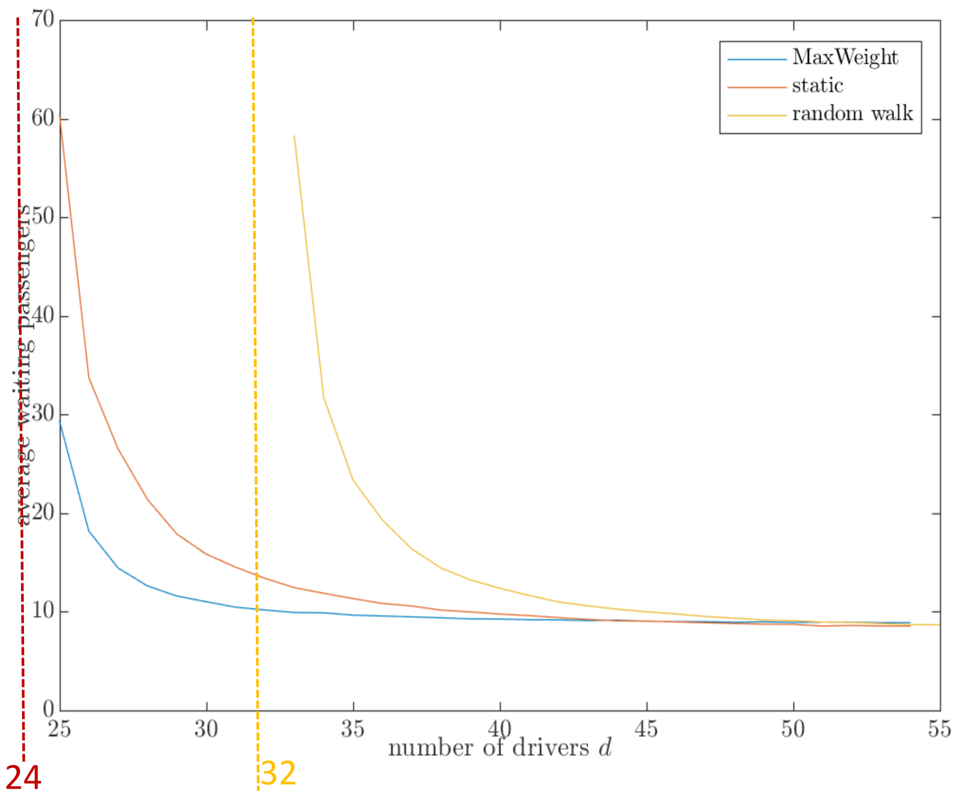


Numerical example

- Input rate asymmetry: large gains by incentive policies



(a) Unbalanced demand: $\lambda_1 = 3, \lambda_4 = 5$



(b) Balanced demand: $\lambda_1 = \lambda_4 = 4$

Conclusions

- A model of ride-hailing with passenger queueing
- Obtained the set of possible passenger service rates under various set of policies
  - Nonunique service rates: if unstable, rational drivers may get trapped into multiple recurrent classes
  - If stable, uniquely defined service rates in simple systems
- Uber type surge pricing interpreted as MaxWeight scheduling
  - Throughput optimal
  - Results to lower delays than the stability-optimal static policy in simulations

Open issues

- Uniqueness of driver response for any set of rewards
- Delay performance of various policies
- Is MaxWeight stability achieving if drivers are more myopic, or less exact
- Long-run equilibrium?

The End

Extension to Markov Decision Processes

- *Do reward incentives exist such that a unique desired behavior is effected?*
- Inverse problem of an MDP
  - Given:
    - finite state and action MDP
    - Desired behavior: π_{ia} = probability of being in state i and taking action a, for all i, a
  - Determine reward r_{ia} given in state i for taking action a, for all i, a *such that an optimal policy has the desired behavior*
- Issue: multiple optimal policies exist if desired behavior is nondeterministic
- Random rewarding: reward with probability ϕ_i, else no reward

Conjecture: there exist rewards r and reward probabilities ϕ such that the only optimal policy is the one implied by the desired behavior

Subtleties

- In the fixed mobility case, fictitious service increases the stability region as the system can use it to perform routing
  - In the flexible mobility stability is not facilitated since routing is optimized anyway.
- MaxWeight does not require any special selection between optimal reward-policies, if multiple exist.
- In the flexible mobility case, the use of b_i's in the definition of the stability region (slide 24) is superfluous.
  - So we could have removed b_i's and note that f_{ij} is the rate of non serving drivers (free or busy).
- We do not need to make any assumption for separation of timescales. This is OK since we are interested in stability. Separation between queue state evolution and the driver control problem obtains at the fluid limit – which is relevant for stability.
- When is the assumption that the drivers are not strategic, nor anticipate any queue state changes is justified?

Reward-optimizing policies

- A passenger pick up from i generates reward at rate r_i for the duration of transport
  - Reward is generated also for fictitious service
- Equivalently:

$$
\begin{aligned}
d\gamma = \max \sum_{i,j} \frac{r_i}{\mu_i} b_i \\
\text{s.t. } b_i + \sum_j f_{ij} = \sum_j b_j q_{ji} + \sum_j f_{ji}, \forall i \\
\sum_i \frac{b_i}{\mu_i} + \sum_{i,j} \frac{f_{ij}}{\mu_{ij}} = d \\
\text{over } b_i, f_{ij} \geq 0, i, j = 1, \ldots, r
\end{aligned}
$$

Stability region

- Notice that a stable process (**not necessarily static**) satisfies:
  - $\sum_j f_{ij} + b_i = \sum_j f_{ji} + \sum_j b_j q_{ji}, \forall i$
  - $\sum_i \frac{b_i}{\mu_i} + \sum_{i,j} \frac{f_{ij}}{\mu_{ij}} = d$

  for some $b_i \geq \lambda_i, f_i \geq 0$ *for all* i.

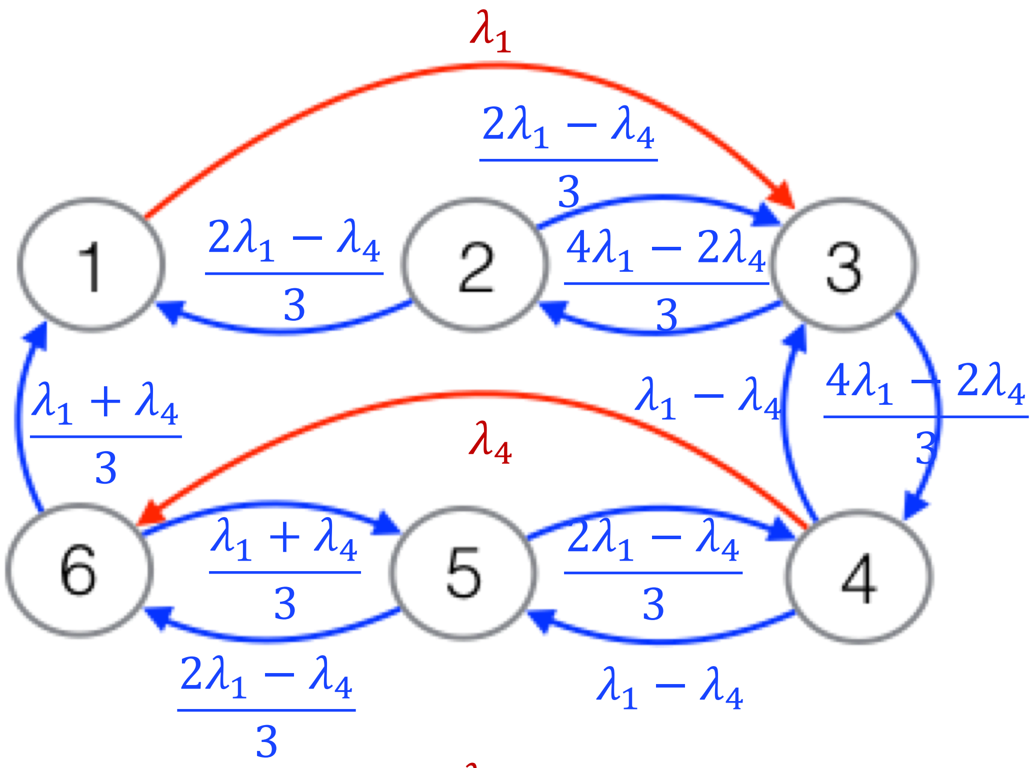
Let Λ = set of (λ_i) which satisfy

  - b_i: rate of drivers becoming busy in region i (incl. fictitious service)
  - f_{ij}: rate of drivers idling in region i and next move to j
- For any $\lambda \in \Lambda$ there exists a ***static policy*** which is stable under appropriate initial placement of drivers. (Theorem 1)
- ***Corollary:*** *A stable policy for arrival rates λ exists if and only if $\lambda \in \Lambda$.*
  - *Dynamic policies are as good as static in terms of stability.*
  - Λ is the *stability region*.

Example

- Assume $\lambda_1 > \lambda_4$
- Idle drivers follow a symmetric random walk:

$$10(\lambda_1 + \lambda_4) - 12\min(\lambda_1, \lambda_4) \leq d$$



- Optimal static policy:
  - Obtained by solving the transshipment problem

$$2(\lambda_1 + \lambda_4) + 2\max(\lambda_1, \lambda_4) \leq d$$