Learning While Repositioning in On-Demand Inventory Sharing Networks

Hansheng Jiang (University of Toronto)

Joint work with
Shunan Jiang (UC Berkeley), Max Shen (HKU), and Chunlin Sun (Stanford)

POMS-HK 2024

Vehicle Sharing Networks

Vehicle Sharing Networks

Features

- On-demand: customers reserve a vehicle when they want
- One-way: rent from one location and return the vehicle to *any other* location in the service network
- Examples: bikes, scooters, cars

Vehicle Sharing Networks

Features

- On-demand: customers reserve a vehicle when they want
- One-way: rent from one location and return the vehicle to *any other* location in the service network
- Examples: bikes, scooters, cars

Source: Generated by Midjourney

Vehicle Sharing Networks

Features

- On-demand: customers reserve a vehicle when they want
- One-way: rent from one location and return the vehicle to *any other* location in the service network
- Examples: bikes, scooters, cars

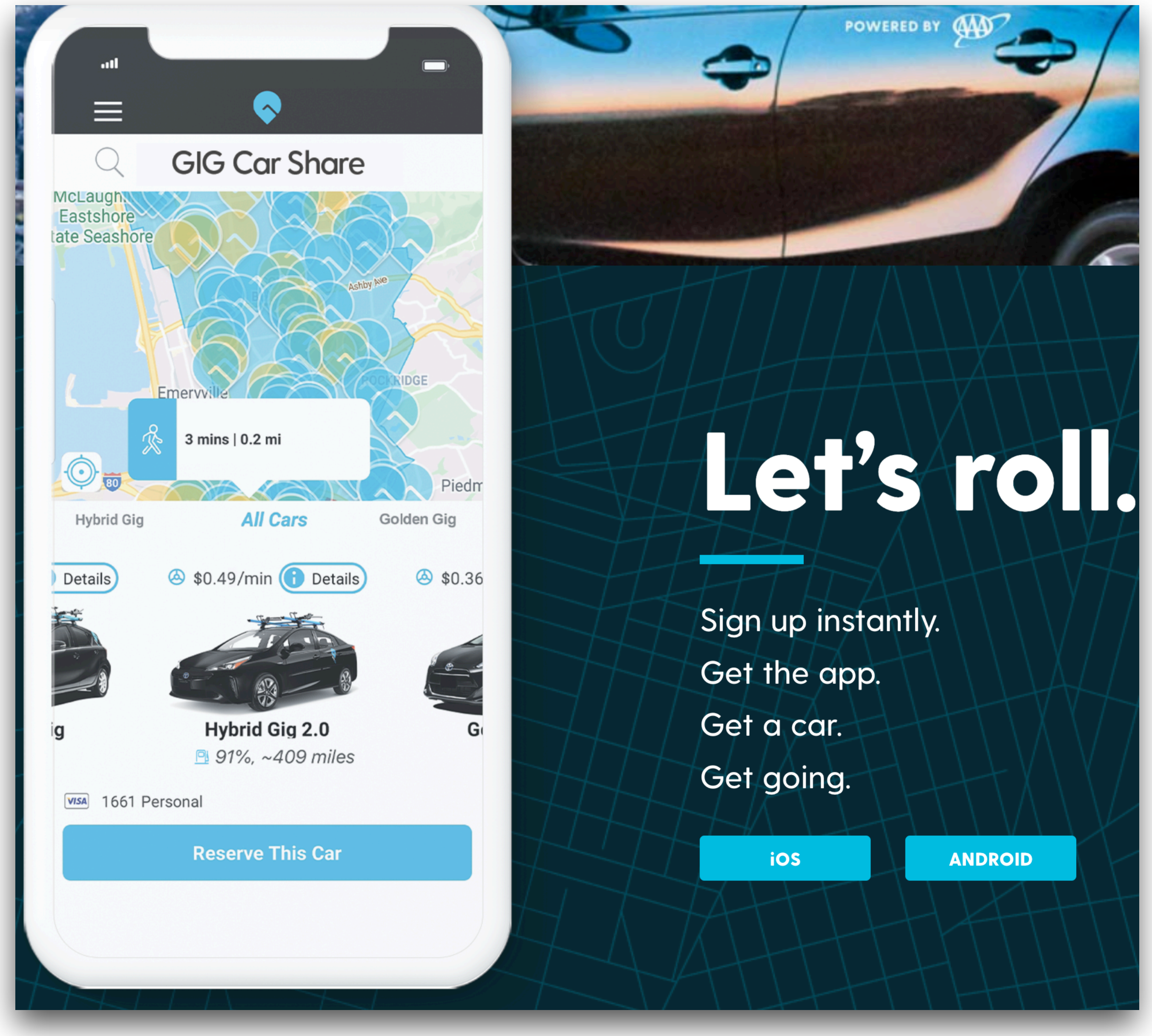
Benefits

- Increased flexibility and convenience for customers
- Competitive transportation costs for customers
- Environmental friendly
  - May reduce overall vehicle ownerships and produce less carbon emissions
  - Help to promote adoption of EVs with cleaner energy

Source: Generated by Midjourney

Emerging Platforms and Programs

Emerging Platforms and Programs

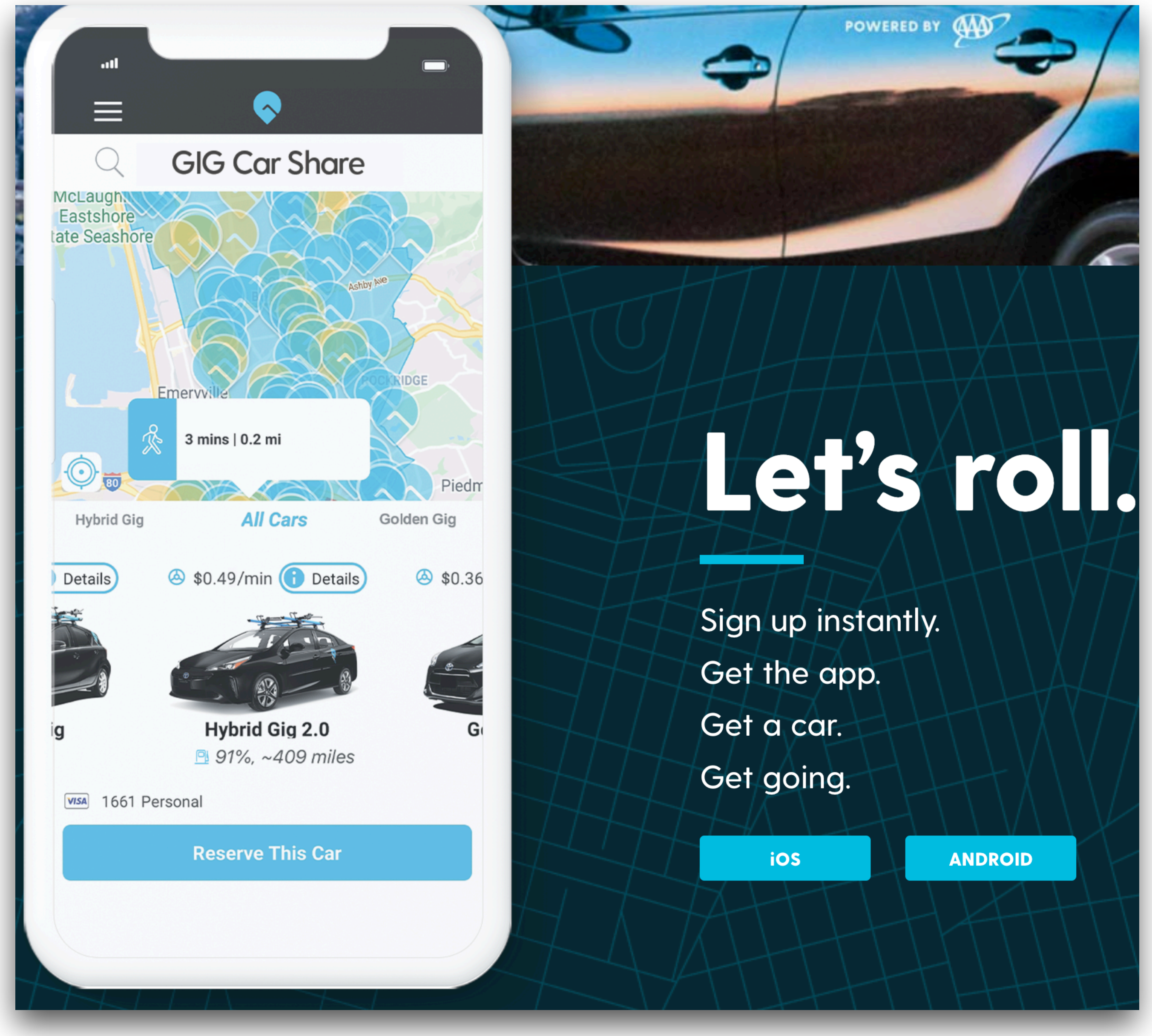


Source: gigcarshare.com

"GIG Car Share: Carsharing service in parts of the San Francisco Bay Area, Sacramento, and Seattle, created by A3 Ventures

The company operates a fleet of Toyota Prius Hybrid vehicles and all-electric Chevrolet Bolts. It offers one-way point-to-point rentals."

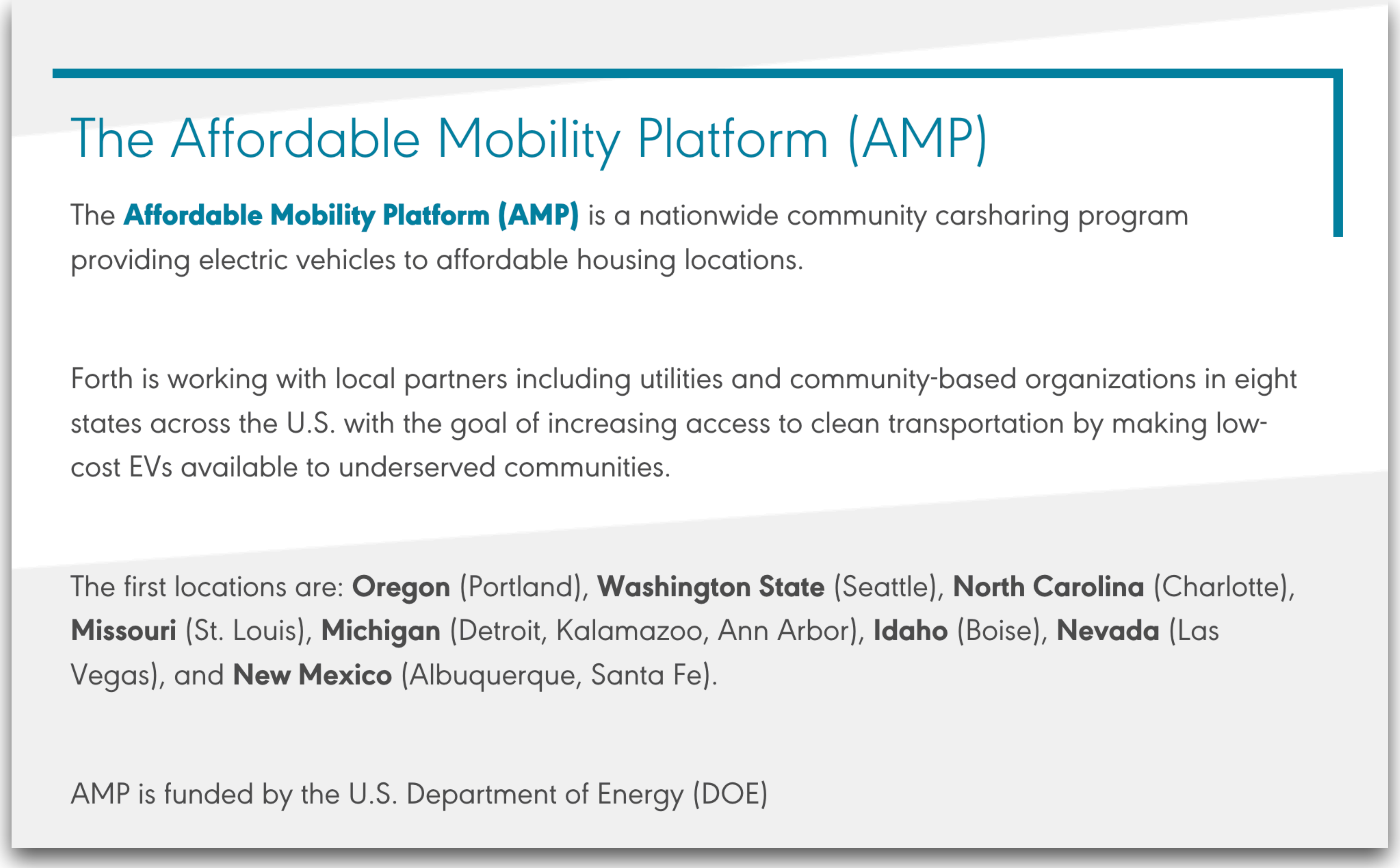
Emerging Platforms and Programs



Source: gigcarshare.com

"GIG Car Share: Carsharing service in parts of the San Francisco Bay Area, Sacramento, and Seattle, created by A3 Ventures

The company operates a fleet of Toyota Prius Hybrid vehicles and all-electric Chevrolet Bolts. It offers one-way point-to-point rentals."



The Affordable Mobility Platform (AMP)

The **Affordable Mobility Platform (AMP)** is a nationwide community carsharing program providing electric vehicles to affordable housing locations.

Forth is working with local partners including utilities and community-based organizations in eight states across the U.S. with the goal of increasing access to clean transportation by making low-cost EVs available to underserved communities.

The first locations are: **Oregon** (Portland), **Washington State** (Seattle), **North Carolina** (Charlotte), **Missouri** (St. Louis), **Michigan** (Detroit, Kalamazoo, Ann Arbor), **Idaho** (Boise), **Nevada** (Las Vegas), and **New Mexico** (Albuquerque, Santa Fe).

AMP is funded by the U.S. Department of Energy (DOE)

Source: forthmobility.org/community-carsharing

Nationwide community carsharing program addressing lack of public transportation and providing cleaner transportation option for low-income community

Motivation

Motivation

Numerous **operational challenges** of vehicle sharing networks

- Service region design
- Fleet sizing
- Trip pricing
- Infrastructure planning

……

Motivation

Numerous **operational challenges** of vehicle sharing networks

- Service region design
- Fleet sizing
- Trip pricing
- Infrastructure planning

……

Focus of this talk: **Inventory Repositioning**

- Lost demand due to lack of vehicles in high utilization zone
- Low utilization zone with oversupply of vehicles

Motivation

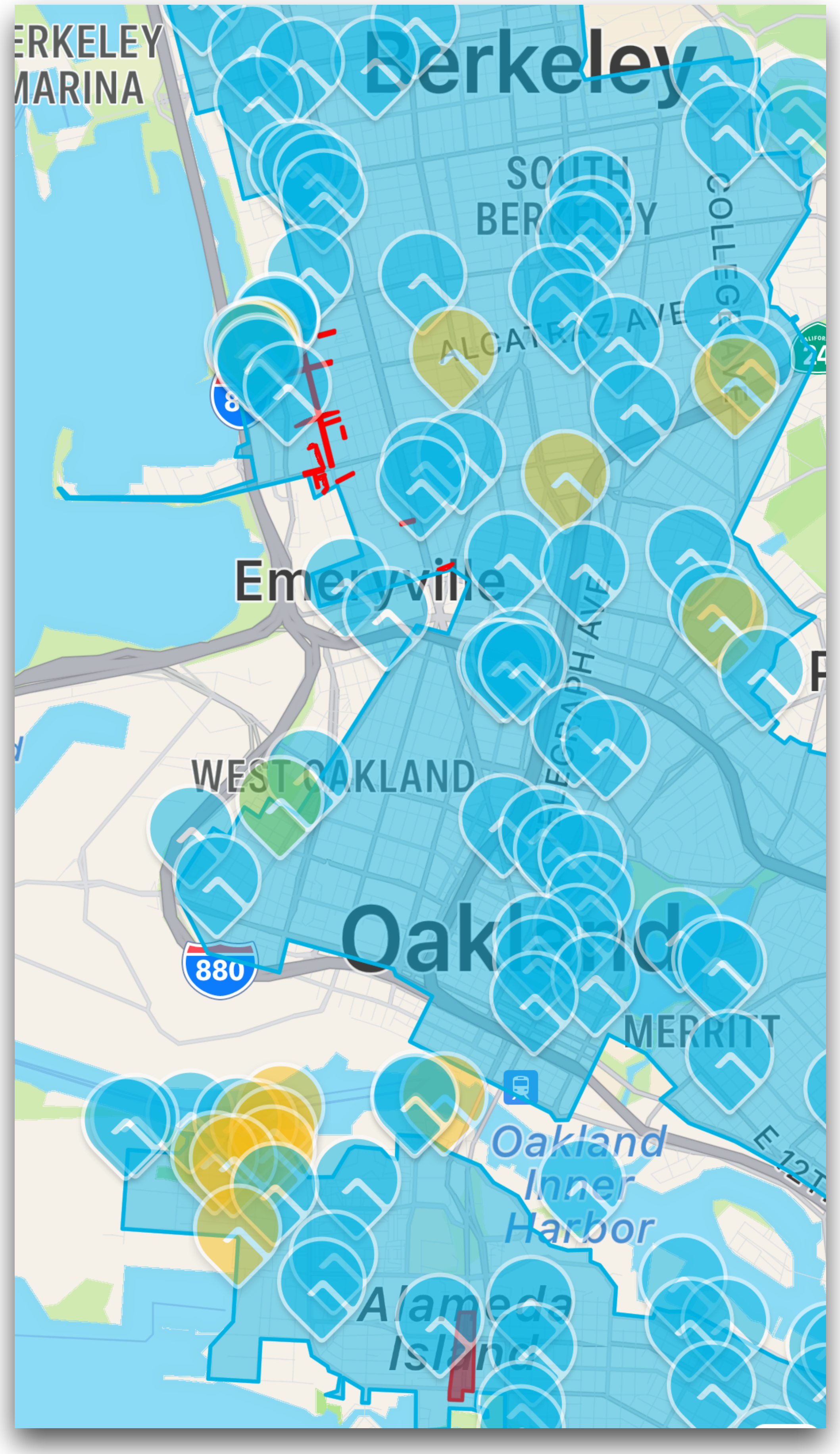
Numerous **operational challenges** of vehicle sharing networks

- Service region design
- Fleet sizing
- Trip pricing
- Infrastructure planning

……

Focus of this talk: **Inventory Repositioning**

- Lost demand due to lack of vehicles in high utilization zone
- Low utilization zone with oversupply of vehicles



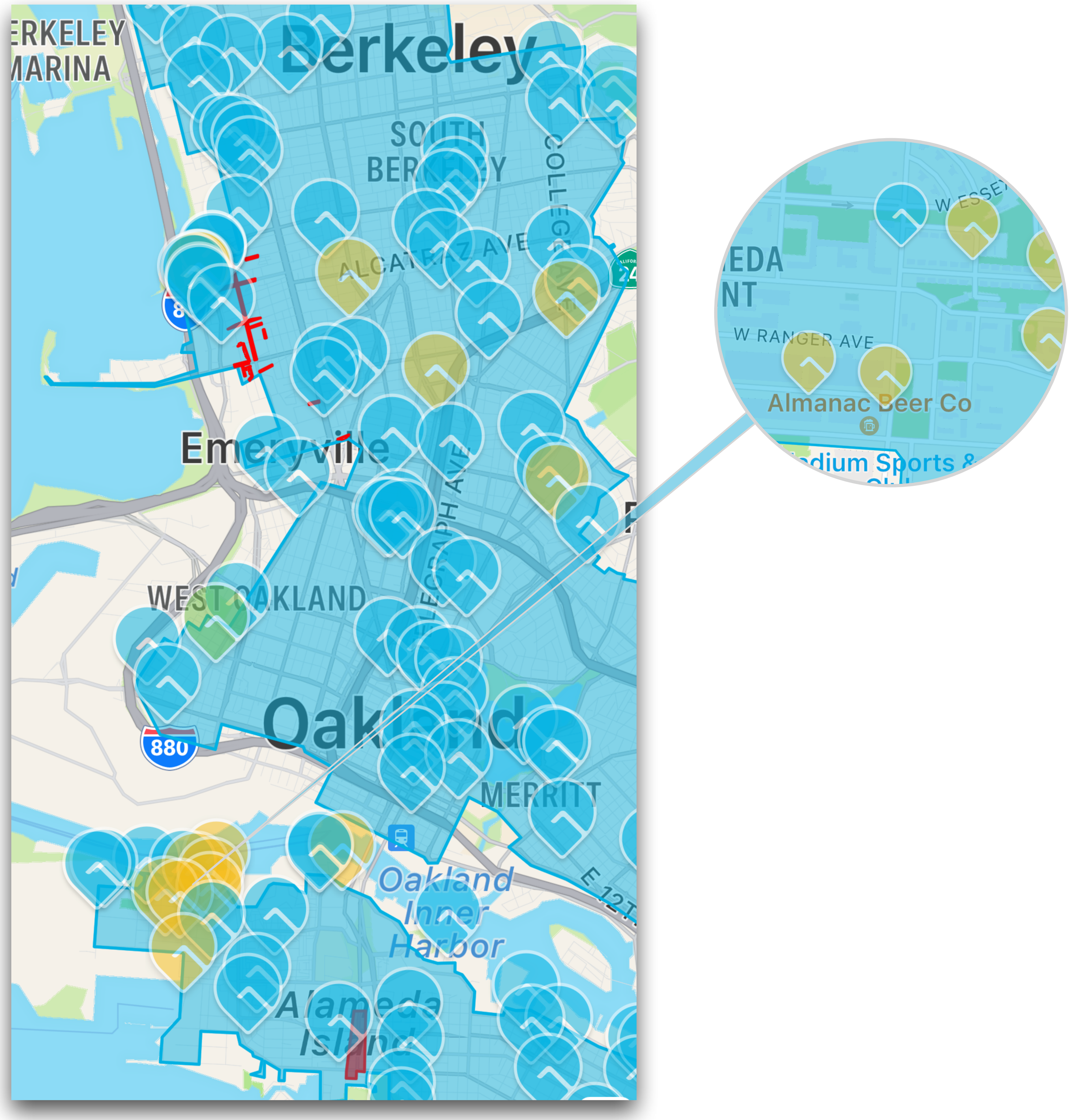
Screenshot of GIG Car Share App

Motivation

Numerous **operational challenges** of vehicle sharing networks

- Service region design
- Fleet sizing
- Trip pricing
- Infrastructure planning

……



Screenshot of GIG Car Share App

Focus of this talk: **Inventory Repositioning**

- Lost demand due to lack of vehicles in high utilization zone
- Low utilization zone with oversupply of vehicles

Matching Supply with Demand in Network

Matching Supply with Demand in Network

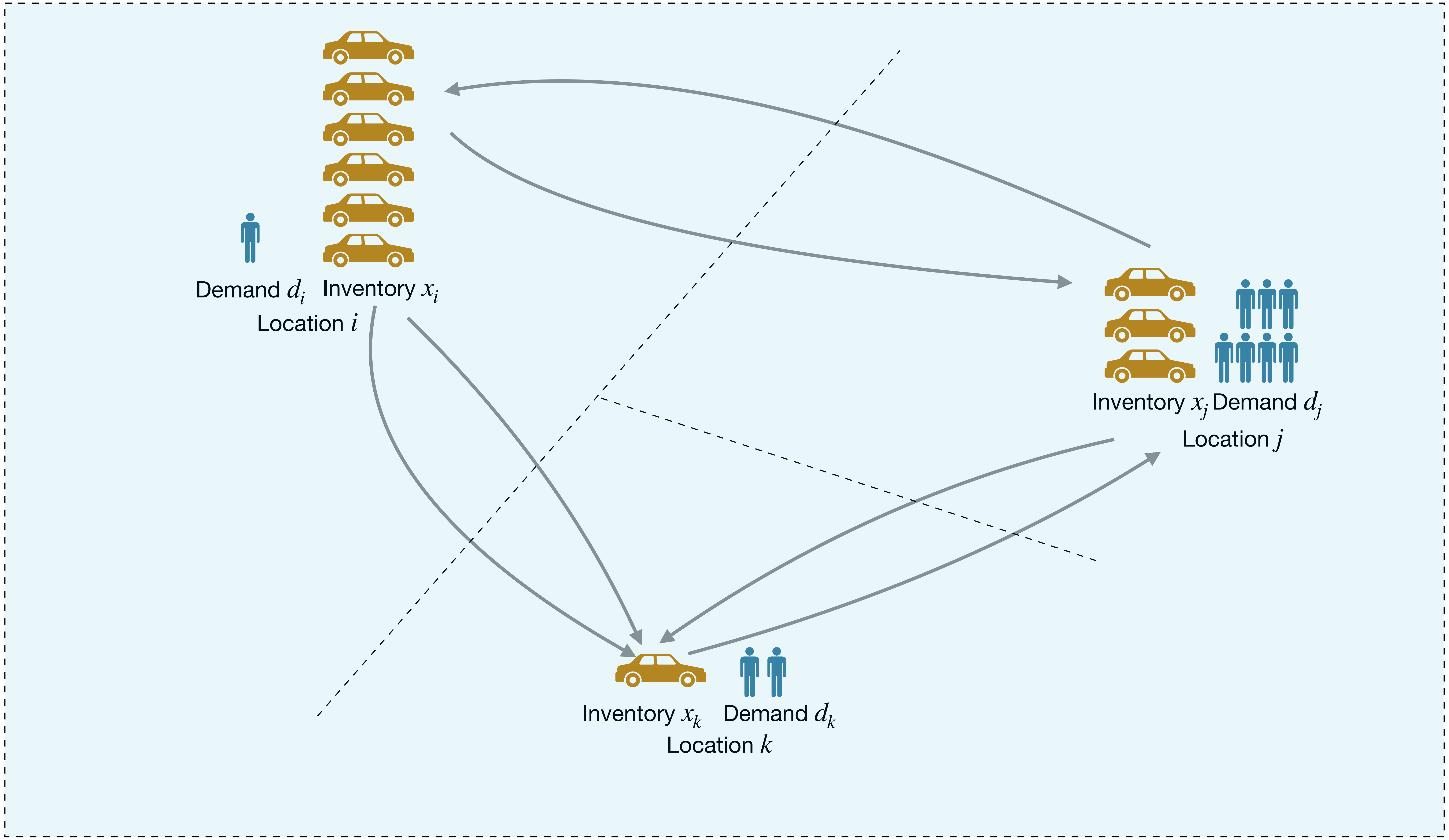


Illustration of 3 locations in a n-location service region

Inventory Dynamics as MDP

Inventory Dynamics as MDP

Markov Decision Process

Inventory Dynamics as MDP

Markov Decision Process

At period $t = 1, 2, \ldots$

Inventory Dynamics as MDP

Markov Decision Process

At period $t = 1,2,\ldots$

I) Service provider reviews the current inventory level $\boldsymbol{x}_t$ (State)

Inventory Dynamics as MDP

Markov Decision Process

At period $t = 1,2,\ldots$

I) Service provider reviews the current inventory level $\boldsymbol{x}_t$ (State)

II) Service provider makes a decision on the target repositioning inventory level $\boldsymbol{y}_t$ (Policy)

Inventory Dynamics as MDP

Markov Decision Process

At period $t = 1,2,\ldots$

I) Service provider reviews the current inventory level $\boldsymbol{x}_t$ (State)

II) Service provider makes a decision on the target repositioning inventory level $\boldsymbol{y}_t$ (Policy)

- Repositioning policy:

Inventory Dynamics as MDP

Markov Decision Process

At period $t = 1,2,\ldots$

I) Service provider reviews the current inventory level $\boldsymbol{x}_t$ (State)

II) Service provider makes a decision on the target repositioning inventory level $\boldsymbol{y}_t$ (Policy)

- Repositioning policy:

$$\boldsymbol{x}_t = (x_{t,1}, \ldots, x_{t,n}) \xrightarrow{\text{policy } \pi} \boldsymbol{y}_t = (y_{t,1}, \ldots, y_{t,n})$$

Inventory Dynamics as MDP

Markov Decision Process

At period $t = 1,2,\ldots$

I) Service provider reviews the current inventory level $\boldsymbol{x}_t$ (State)

II) Service provider makes a decision on the target repositioning inventory level $\boldsymbol{y}_t$ (Policy)

- Repositioning policy:

$$\boldsymbol{x}_t = (x_{t,1}, \ldots, x_{t,n}) \xrightarrow{\text{policy } \pi} \boldsymbol{y}_t = (y_{t,1}, \ldots, y_{t,n})$$

Inventory Dynamics as MDP

Markov Decision Process

At period $t = 1,2,\ldots$

I) Service provider reviews the current inventory level $\boldsymbol{x}_t$ (State)

II) Service provider makes a decision on the target repositioning inventory level $\boldsymbol{y}_t$ (Policy)

- Repositioning policy:

$$\boldsymbol{x}_t = (x_{t,1}, \ldots, x_{t,n}) \xrightarrow{\text{policy } \pi} \boldsymbol{y}_t = (y_{t,1}, \ldots, y_{t,n})$$

III) Rental trips by customers are realized, and inventory level moves to a new level $\boldsymbol{x}_{t+1}$

Inventory Dynamics as MDP

Markov Decision Process

At period $t = 1,2,\ldots$

I) Service provider reviews the current inventory level $\boldsymbol{x}_t$ (State)

II) Service provider makes a decision on the target repositioning inventory level $\boldsymbol{y}_t$ (Policy)

- Repositioning policy:

$$\boldsymbol{x}_t = (x_{t,1}, \ldots, x_{t,n}) \xrightarrow{\text{policy } \pi} \boldsymbol{y}_t = (y_{t,1}, \ldots, y_{t,n})$$

III) Rental trips by customers are realized, and inventory level moves to a new level $\boldsymbol{x}_{t+1}$

- Rental trips

Inventory Dynamics as MDP

Markov Decision Process

At period $t = 1,2,\ldots$

I) Service provider reviews the current inventory level $\boldsymbol{x}_t$ (State)

II) Service provider makes a decision on the target repositioning inventory level $\boldsymbol{y}_t$ (Policy)

- Repositioning policy:

$$\boldsymbol{x}_t = (x_{t,1}, \ldots, x_{t,n}) \xrightarrow{\text{policy } \pi} \boldsymbol{y}_t = (y_{t,1}, \ldots, y_{t,n})$$

III) Rental trips by customers are realized, and inventory level moves to a new level $\boldsymbol{x}_{t+1}$

- Rental trips
  - Censored demand $\min(\boldsymbol{d}_t, \boldsymbol{y}_t)$

Inventory Dynamics as MDP

Markov Decision Process

At period $t = 1,2,\ldots$

I) Service provider reviews the current inventory level $\boldsymbol{x}_t$ (State)

II) Service provider makes a decision on the target repositioning inventory level $\boldsymbol{y}_t$ (Policy)

- Repositioning policy:

$$\boldsymbol{x}_t = (x_{t,1}, \ldots, x_{t,n}) \xrightarrow{\text{policy } \pi} \boldsymbol{y}_t = (y_{t,1}, \ldots, y_{t,n})$$

III) Rental trips by customers are realized, and inventory level moves to a new level $\boldsymbol{x}_{t+1}$

- Rental trips
  - Censored demand $\min(\boldsymbol{d}_t, \boldsymbol{y}_t)$
  - Origin-to-destination matrix for vehicles returning $\boldsymbol{P}_t$

Inventory Dynamics as MDP

Markov Decision Process

At period $t = 1,2,\ldots$

I) Service provider reviews the current inventory level $\boldsymbol{x}_t$ (State)

II) Service provider makes a decision on the target repositioning inventory level $\boldsymbol{y}_t$ (Policy)

- Repositioning policy:

$$\boldsymbol{x}_t = (x_{t,1}, \ldots, x_{t,n}) \xrightarrow{\text{policy } \pi} \boldsymbol{y}_t = (y_{t,1}, \ldots, y_{t,n})$$

III) Rental trips by customers are realized, and inventory level moves to a new level $\boldsymbol{x}_{t+1}$

- Rental trips
    - Censored demand $\min(\boldsymbol{d}_t, \boldsymbol{y}_t)$
    - Origin-to-destination matrix for vehicles returning $\boldsymbol{P}_t$
- State transition:

Inventory Dynamics as MDP

Markov Decision Process

At period $t = 1,2,\ldots$

I) Service provider reviews the current inventory level $\boldsymbol{x}_t$ (State)

II) Service provider makes a decision on the target repositioning inventory level $\boldsymbol{y}_t$ (Policy)

- Repositioning policy:

$$\boldsymbol{x}_t = (x_{t,1}, \ldots, x_{t,n}) \xrightarrow{\text{policy } \pi} \boldsymbol{y}_t = (y_{t,1}, \ldots, y_{t,n})$$

III) Rental trips by customers are realized, and inventory level moves to a new level $\boldsymbol{x}_{t+1}$

- Rental trips
  - Censored demand $\min(\boldsymbol{d}_t, \boldsymbol{y}_t)$
  - Origin-to-destination matrix for vehicles returning $\boldsymbol{P}_t$
- State transition:

$$\boldsymbol{x}_{t+1} = (\boldsymbol{y}_t - \boldsymbol{d}_t)^+ + \boldsymbol{P}^T \min(\boldsymbol{y}_t, \boldsymbol{d}_t)$$

Objective

Objective

Single-period cost of policy π

Total cost C_t^π = Repositioning cost M_t^π + Lost sales cost L_t^π

Objective

Single-period cost of policy π

Total cost C_t^π = Repositioning cost M_t^π + Lost sales cost L_t^π

$$M_t = \min \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} \cdot \xi_{ij}$$

Obtained by solving minimum cost flow problem

$$\text{s.t.} \sum_{i=1}^{n} \xi_{ij} - \sum_{k=1}^{n} \xi_{jk} = y_{t,j} - x_{t,j}$$

$$L_t = \sum_{i} \sum_{j} l_{ij} \cdot P_{ij}(d_i - y_i)^+$$

P_{ij} origin-to-destination probability

Objective

Single-period cost of policy π

Total cost C_t^π = Repositioning cost M_t^π + Lost sales cost L_t^π

$$M_t = \min \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} \cdot \xi_{ij}$$

Obtained by solving minimum cost flow problem

$$\text{s.t.} \sum_{i=1}^{n} \xi_{ij} - \sum_{k=1}^{n} \xi_{jk} = y_{t,j} - x_{t,j}$$

$$L_t = \sum_i \sum_j l_{ij} \cdot P_{ij}(d_i - y_i)^+$$

P_{ij} origin-to-destination probability

Long-run average cost of policy π

$$\lambda^\pi = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}[C_t^\pi]$$

Designing Repositioning Policy

Designing Repositioning Policy

Optimal Policy

Designing Repositioning Policy

Optimal Policy

$$\min_{\pi} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}[C_t^{\pi}] \,, T \rightarrow \infty$$

- Optimal policy is computationally expensive in general even when the demand distribution is known or fully observed

Designing Repositioning Policy

Optimal Policy

$$\min_{\pi} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}[C_t^{\pi}]\,,\, T \to \infty$$

- Optimal policy is computationally expensive in general even when the demand distribution is known or fully observed

Designing Repositioning Policy

Optimal Policy

$$\min_{\pi} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}[C_t^{\pi}]\,, T \to \infty$$

- Optimal policy is computationally expensive in general even when the demand distribution is known or fully observed

Base-Stock Repositioning Policy

- **Definition:** Repositioning to base-stock level $\boldsymbol{S} = (S_1, \ldots, S_n)$ regardless of the current state $\boldsymbol{x}_t$

Designing Repositioning Policy

Optimal Policy

$$\min_{\pi} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}[C_t^{\pi}] \,,\, T \to \infty$$

- Optimal policy is computationally expensive in general even when the demand distribution is known or fully observed

Base-Stock Repositioning Policy

- **Definition:** Repositioning to base-stock level $\boldsymbol{S} = (S_1, \ldots, S_n)$ regardless of the current state $\boldsymbol{x}_t$

Best Base-Stock Repositioning Policy

Designing Repositioning Policy

Optimal Policy

$$\min_{\pi} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}[C_t^{\pi}]\,,\, T \to \infty$$

- Optimal policy is computationally expensive in general even when the demand distribution is known or fully observed

Base-Stock Repositioning Policy

- **Definition:** Repositioning to base-stock level $\boldsymbol{S} = (S_1, \ldots, S_n)$ regardless of the current state $\boldsymbol{x}_t$

Best Base-Stock Repositioning Policy

$$\mathbf{S}^{\star} \in \arg \min_{\mathbf{S} \in \Delta_{n-1}} \limsup_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}^{\mathbf{S}}[C_t]$$

Asymptotic Optimality of Base-Stock Policy

Asymptotic Optimality of Base-Stock Policy

Theorem (*informal*) The ratio of the *optimal* base-stock repositioning policy's long-run average cost to the optimal repositioning policy's long-run average cost approaches 1 when the ratio of unit lost sales cost to unit repositioning cost $l_{ij}/c_{ij} \to \infty$.

Asymptotic Optimality of Base-Stock Policy

Theorem (*informal*) The ratio of the *optimal* base-stock repositioning policy's long-run average cost to the optimal repositioning policy's long-run average cost approaches 1 when the ratio of unit lost sales cost to unit repositioning cost $l_{ij}/c_{ij} \to \infty$.

Intuition

Repositioning can be done in bulk; Minimizing user dissatisfication; Need for market growth

Asymptotic Optimality of Base-Stock Policy

Theorem (*informal*) The ratio of the *optimal* base-stock repositioning policy's long-run average cost to the optimal repositioning policy's long-run average cost approaches 1 when the ratio of unit lost sales cost to unit repositioning cost $l_{ij}/c_{ij} \to \infty$.

Intuition

Repositioning can be done in bulk; Minimizing user dissatisfication; Need for market growth

Theorem (*informal*) The ratio of the *optimal* base-stock repositioning policy's long-run average cost to the optimal repositioning policy's long-run average cost approaches 1 when the number of locations n in the network goes to ∞ .

Asymptotic Optimality of Base-Stock Policy

Theorem (*informal*) The ratio of the *optimal* base-stock repositioning policy's long-run average cost to the optimal repositioning policy's long-run average cost approaches 1 when the ratio of unit lost sales cost to unit repositioning cost $l_{ij}/c_{ij} \to \infty$.

Intuition

Repositioning can be done in bulk; Minimizing user dissatisfication; Need for market growth

Theorem (*informal*) The ratio of the *optimal* base-stock repositioning policy's long-run average cost to the optimal repositioning policy's long-run average cost approaches 1 when the number of locations n in the network goes to ∞ .

Intuition

Lost sales cost occurred individually at each location — the opposite of "risk pooling"

Learning Best Base-Stock Policy

Learning Best Base-Stock Policy

Performance Metric The regret compared with the base-stock repositioning policy incurred by algorithm A with optimal base-stock level $S^\star$

$$\mathrm{Regret}(\mathrm{A}, T) = \sum_{t=1}^{T} \mathbb{E}[C_t^{\mathrm{A}}] - \sum_{t=1}^{T} \mathbb{E}[C_t^{S^\star}]$$

Learning Best Base-Stock Policy

Performance Metric The regret compared with the base-stock repositioning policy incurred by algorithm A with optimal base-stock level $S^\star$

$$\mathrm{Regret}(\mathrm{A}, T) = \sum_{t=1}^{T} \mathbb{E}[C_t^{\mathrm{A}}] - \sum_{t=1}^{T} \mathbb{E}[C_t^{S^\star}]$$

Bandit Learning Perspective

- Treat each base-stock repositioning policy as an arm
- The reward of each arm is negative long-run average cost

Learning Best Base-Stock Policy

Performance Metric The regret compared with the base-stock repositioning policy incurred by algorithm A with optimal base-stock level $S^\star$

$$\mathrm{Regret}(\mathrm{A}, T) = \sum_{t=1}^{T} \mathbb{E}[C_t^{\mathrm{A}}] - \sum_{t=1}^{T} \mathbb{E}[C_t^{S^\star}]$$

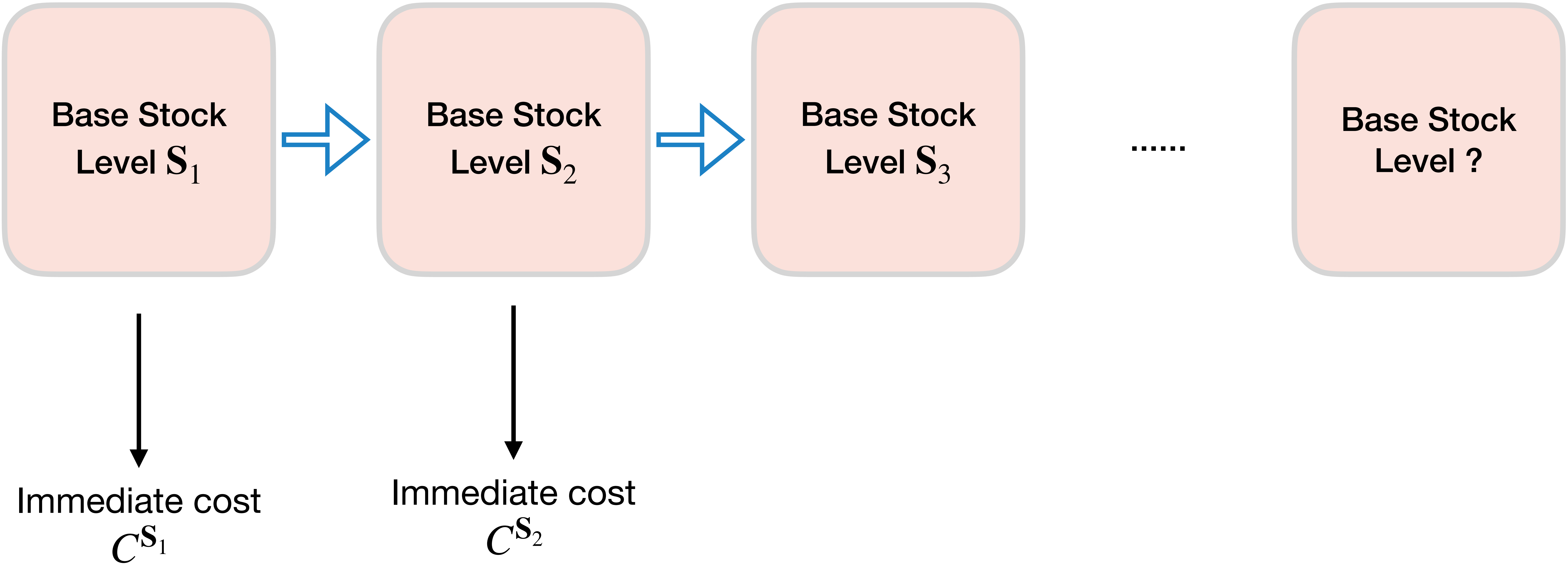
Bandit Learning Perspective

- Treat each base-stock repositioning policy as an arm
- The reward of each arm is negative long-run average cost

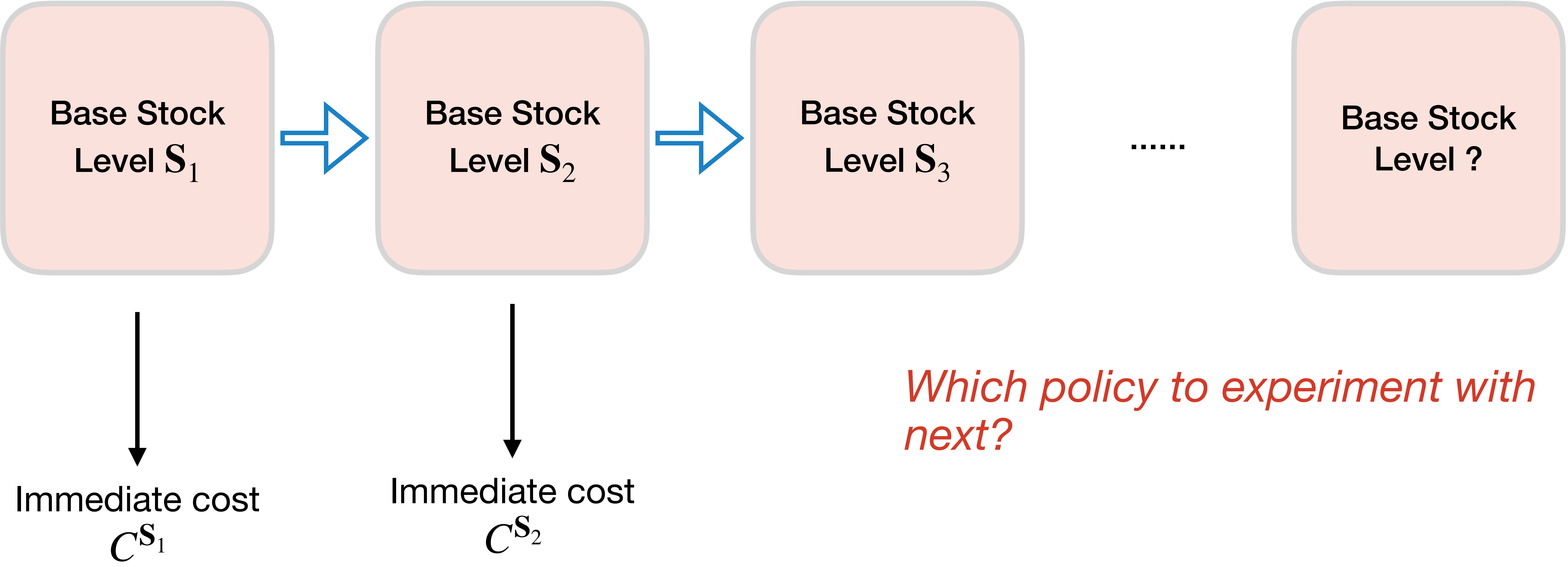
Difficulties

- Only censored demand is known
- Reward is not immediately accessible and only partially observed
- Randomness in both demand arriving and vehicle returning
- Curse of dimensionality in a network with multiple locations

Learning While Repositioning



Learning While Repositioning



Learning While Repositioning

Base Stock Level S_1 → Base Stock Level S_2 → Base Stock Level S_3 …… Base Stock Level ?

Immediate cost C^{S_1}

Immediate cost C^{S_2}

Which policy to experiment with next?

Compared with classical MAB literature:
demand is censored
immediate single period cots $\neq$ long run average cost
potential policies are not finite
….

Lipschitz Bandits-based Repositioning

`LipBR` Algorithm

Lipschitz Bandits-based Repositioning

LipBR Algorithm

Algorithm Design Idea

- Establish Lipschitz property of the long-run average cost wrt policy
- Discretize the policy space Δ_{n-1} by covering, and bound the covering number by $O(\epsilon^{1-n})$ for accuracy ϵ
- Concentration inequalities of single period costs versus long-run average costs
- Monitor pseudo costs $\tilde{C}$ in regret definition to address unobservable lost sales cost
- Regret $\approx \sqrt{KT} + K\epsilon$, where $K = O(\epsilon^{1-n})$ and $\epsilon = O(T^{-1/(n+1)})$

Lipschitz Bandits-based Repositioning

LipBR Algorithm

Algorithm Design Idea

- Establish Lipschitz property of the long-run average cost wrt policy
- Discretize the policy space Δ_{n-1} by covering, and bound the covering number by $O(\epsilon^{1-n})$ for accuracy ϵ
- Concentration inequalities of single period costs versus long-run average costs
- Monitor pseudo costs $\tilde{C}$ in regret definition to address unobservable lost sales cost
- Regret $\approx \sqrt{KT} + K\epsilon$, where $K = O(\epsilon^{1-n})$ and $\epsilon = O(T^{-1/(n+1)})$

Theorem (*informal*) The regret of the **LipBR** algorithm against the optimal base-stock policy is upper bounded by $\tilde{O}(T^{\frac{n}{n+1}})$.

Lipschitz Bandits-based Repositioning

LipBR Algorithm

Algorithm Design Idea

- Establish Lipschitz property of the long-run average cost wrt policy
- Discretize the policy space Δ_{n-1} by covering, and bound the covering number by $O(\epsilon^{1-n})$ for accuracy ϵ
- Concentration inequalities of single period costs versus long-run average costs
- Monitor pseudo costs $\tilde{C}$ in regret definition to address unobservable lost sales cost
- Regret $\approx \sqrt{KT} + K\epsilon$, where $K = O(\epsilon^{1-n})$ and $\epsilon = O(T^{-1/(n+1)})$

Theorem (*informal*) The regret of the **LipBR** algorithm against the optimal base-stock policy is upper bounded by $\tilde{O}(T^{\frac{n}{n+1}})$.

Can we bypass the curse of dimensionality and remove the power dependence on n?

Results by Solving the Offline Problem

Results by Solving the Offline Problem

Offline Optimization Problem

Results by Solving the Offline Problem

Offline Optimization Problem

$$\min_{\mathbf{s}\in\Delta_{n-1}} \frac{1}{T}\sum_{t=1}^{T} \tilde{C}^{\mathbf{s}}$$

Results by Solving the Offline Problem

Offline Optimization Problem

$$\min_{\mathbf{s} \in \Delta_{n-1}} \frac{1}{T} \sum_{t=1}^{T} \tilde{C}^{\mathbf{s}}$$

Results by Solving the Offline Problem

Offline Optimization Problem

$$\min_{\mathbf{s}\in\Delta_{n-1}} \frac{1}{T}\sum_{t=1}^{T} \tilde{C}^{\mathbf{s}}$$

Results by Solving the Offline Problem

Offline Optimization Problem

$$\min_{\mathbf{s}\in\Delta_{n-1}} \frac{1}{T}\sum_{t=1}^{T} \tilde{C}^{\mathbf{s}}$$

Results by Solving the Offline Problem

Offline Optimization Problem

$$\min_{\mathbf{s}\in\Delta_{n-1}} \frac{1}{T}\sum_{t=1}^{T} \tilde{C}\mathbf{s}$$

Generalization bound Uniform convergence over Δ_{n-1}

Results by Solving the Offline Problem

Offline Optimization Problem

$$\min_{\mathbf{s}\in\Delta_{n-1}} \frac{1}{T}\sum_{t=1}^{T} \tilde{C}^{\mathbf{s}}$$

Generalization bound Uniform convergence over Δ_{n-1}

Two Reformulations Tackling NonConvexity

Results by Solving the Offline Problem

Offline Optimization Problem

$$\min_{\mathbf{s} \in \Delta_{n-1}} \frac{1}{T} \sum_{t=1}^{T} \tilde{C}^{\mathbf{s}}$$

Generalization bound Uniform convergence over Δ_{n-1}

Two Reformulations Tackling NonConvexity

- Mixed integer linear programming (MILP) formulation

Results by Solving the Offline Problem

Offline Optimization Problem

$$\min_{\mathbf{s}\in\Delta_{n-1}} \frac{1}{T}\sum_{t=1}^{T} \tilde{C}^{\mathbf{s}}$$

Generalization bound Uniform convergence over Δ_{n-1}

Two Reformulations Tackling NonConvexity

- Mixed integer linear programming (MILP) formulation
- Linear programming (LP) formulation under additional cost assumptions

Results by Solving the Offline Problem

Offline Optimization Problem

$$\min_{\mathbf{s}\in\Delta_{n-1}} \frac{1}{T}\sum_{t=1}^{T} \tilde{C}^{\mathbf{s}}$$

Generalization bound Uniform convergence over Δ_{n-1}

Two Reformulations Tackling NonConvexity

- Mixed integer linear programming (MILP) formulation
- Linear programming (LP) formulation under additional cost assumptions

Results by Solving the Offline Problem

Offline Optimization Problem

$$\min_{\mathbf{s}\in\Delta_{n-1}} \frac{1}{T}\sum_{t=1}^{T} \tilde{C}^{\mathbf{s}}$$

Generalization bound Uniform convergence over Δ_{n-1}

Two Reformulations Tackling NonConvexity

- Mixed integer linear programming (MILP) formulation
- Linear programming (LP) formulation under additional cost assumptions

Theorem (*informal*) Under demand independence assumption, the one-time learning algorithm can achieve $\tilde{O}(T^{\frac{2}{3}})$ regret.

Results by Solving the Offline Problem

Offline Optimization Problem

$$\min_{\mathbf{s}\in\Delta_{n-1}} \frac{1}{T}\sum_{t=1}^{T} \tilde{C}^{\mathbf{s}}$$

Generalization bound Uniform convergence over Δ_{n-1}

Two Reformulations Tackling NonConvexity

- Mixed integer linear programming (MILP) formulation
- Linear programming (LP) formulation under additional cost assumptions

Theorem (*informal*) Under demand independence assumption, the one-time learning algorithm can achieve $\tilde{O}(T^{\frac{2}{3}})$ regret.

Theorem (*informal*) With uncensored demand data, the dynamic learning algorithm can achieve $\tilde{O}(T^{\frac{1}{2}})$ regret.

Regret Analysis of Offline-Based Algorithm

Regret Analysis of Offline-Based Algorithm

Concentration Inequality

With probability at least $1 - \frac{1}{T^2}$, it holds that

$$\sup_{\mathbf{s} \in \Delta_{n-1}} \left| \frac{1}{t} \sum_{s=1}^{t} \widetilde{C}_s^{\mathbf{s}} - \mathbb{E}[\widetilde{C}_1^{\mathbf{s}}] \right| \leq 6n^3 \left(\max_{i,j} c_{ij} + \max_{i,j} l_{ij} \right) \cdot \frac{\sqrt{\log T}}{\sqrt{t}}$$

Regret Analysis of Offline-Based Algorithm

Concentration Inequality

With probability at least $1 - \frac{1}{T^2}$, it holds that

$$\sup_{\mathbf{S}\in\Delta_{n-1}} \left| \frac{1}{t}\sum_{s=1}^{t} \widetilde{C}_s^{\mathbf{S}} - \mathbb{E}[\widetilde{C}_1^{\mathbf{S}}] \right| \le 6n^3 \left(\max_{i,j} c_{ij} + \max_{i,j} l_{ij} \right) \cdot \frac{\sqrt{\log T}}{\sqrt{t}}$$

One time learning $\tilde{O}(T^{\frac{2}{3}})$

Regret Analysis of Offline-Based Algorithm

Concentration Inequality

With probability at least $1 - \frac{1}{T^2}$, it holds that

$$\sup_{\mathbf{S} \in \Delta_{n-1}} \left| \frac{1}{t} \sum_{s=1}^{t} \widetilde{C}_s^{\mathbf{S}} - \mathbb{E}[\widetilde{C}_1^{\mathbf{S}}] \right| \leq 6n^3 \left(\max_{i,j} c_{ij} + \max_{i,j} l_{ij} \right) \cdot \frac{\sqrt{\log T}}{\sqrt{t}}$$

One time learning $\tilde{O}(T^{\frac{2}{3}})$

1. Explore for $nT^{2/3}$ time periods by placing sufficient inventory in n locations respectively

Regret Analysis of Offline-Based Algorithm

Concentration Inequality

With probability at least $1 - \frac{1}{T^2}$, it holds that

$$\sup_{\mathbf{S}\in\Delta_{n-1}} \left| \frac{1}{t}\sum_{s=1}^{t} \widetilde{C}_s^{\mathbf{S}} - \mathbb{E}[\widetilde{C}_1^{\mathbf{S}}] \right| \le 6n^3 \left(\max_{i,j} c_{ij} + \max_{i,j} l_{ij} \right) \cdot \frac{\sqrt{\log T}}{\sqrt{t}}$$

One time learning $\tilde{O}(T^{\frac{2}{3}})$

1. Explore for $nT^{2/3}$ time periods by placing sufficient inventory in n locations respectively
2. Solve the offline problem using data collected in $nT^{2/3}$ time periods

Regret Analysis of Offline-Based Algorithm

Concentration Inequality

With probability at least $1 - \frac{1}{T^2}$, it holds that

$$\sup_{\mathbf{S} \in \Delta_{n-1}} \left| \frac{1}{t} \sum_{s=1}^{t} \widetilde{C}_s^{\mathbf{S}} - \mathbb{E}[\widetilde{C}_1^{\mathbf{S}}] \right| \leq 6n^3 \left(\max_{i,j} c_{ij} + \max_{i,j} l_{ij} \right) \cdot \frac{\sqrt{\log T}}{\sqrt{t}}$$

One time learning $\tilde{O}(T^{\frac{2}{3}})$

1. Explore for $nT^{2/3}$ time periods by placing sufficient inventory in n locations respectively
2. Solve the offline problem using data collected in $nT^{2/3}$ time periods
3. Exploit the policy learned from the offline problem

Regret Analysis of Offline-Based Algorithm

Concentration Inequality

With probability at least $1 - \frac{1}{T^2}$, it holds that

$$\sup_{\mathbf{S} \in \Delta_{n-1}} \left| \frac{1}{t} \sum_{s=1}^{t} \widetilde{C}_s^{\mathbf{S}} - \mathbb{E}[\widetilde{C}_1^{\mathbf{S}}] \right| \leq 6n^3 \left(\max_{i,j} c_{ij} + \max_{i,j} l_{ij} \right) \cdot \frac{\sqrt{\log T}}{\sqrt{t}}$$

One time learning $\tilde{O}(T^{\frac{2}{3}})$

1. Explore for $nT^{2/3}$ time periods by placing sufficient inventory in n locations respectively
2. Solve the offline problem using data collected in $nT^{2/3}$ time periods
3. Exploit the policy learned from the offline problem

Dynamic learning $\tilde{O}(T^{\frac{1}{2}})$ (if demand is uncensored)

- At each period, solve the offline problem and update the policy

Online Learning with Optimal Regret

Algorithm Design

At iteration t

1. Consider the dual optimal solution $\lambda_{t,i}$ to the constraints $w_{t,i} \leq \min\{d_{t,i}, S_i\}$
2. $g_{t,i} = \lambda_i \mathbf{1}_{\{\min\{d_{t,i},S_{t,i}\}=S_{t,i}\}}$ is a sub-gradient
3. Gradient descent $\widetilde{\mathbf{S}}_t = \mathbf{S}_t - \frac{1}{\sqrt{t}}\mathbf{g}_t$
4. Project $\widetilde{\mathbf{S}}_t$ onto Δ_{n-1} to obtain $\mathbf{S}_{t+1}$

Online Learning with Optimal Regret

Algorithm Design

At iteration t

1. Consider the dual optimal solution $\lambda_{t,i}$ to the constraints $w_{t,i} \leq \min\{d_{t,i}, S_i\}$
2. $g_{t,i} = \lambda_i \mathbf{1}_{\{\min\{d_{t,i}, S_{t,i}\} = S_{t,i}\}}$ is a sub-gradient
3. Gradient descent $\widetilde{\mathbf{S}}_t = \mathbf{S}_t - \frac{1}{\sqrt{t}} \mathbf{g}_t$
4. Project $\widetilde{\mathbf{S}}_t$ onto Δ_{n-1} to obtain $\mathbf{S}_{t+1}$

Theorem (*informal*) The online stochastic gradient **OSG** based algorithm achieves a regret of $\tilde{O}(T^{\frac{1}{2}})$ and this rate even holds for adversarial data. This rate matches the theoretical lower bound.

Summary

Summary

Efficient inventory monitoring is critical for successful operations of vehicle sharing systems

Summary

Efficient inventory monitoring is critical for successful operations of vehicle sharing systems

Summary

Efficient inventory monitoring is critical for successful operations of vehicle sharing systems

We establish asymptotic optimality of base-stock repositioning policy and prove near optimal regret bound of learning

Summary

Efficient inventory monitoring is critical for successful operations of vehicle sharing systems

We establish asymptotic optimality of base-stock repositioning policy and prove near optimal regret bound of learning

Summary

Efficient inventory monitoring is critical for successful operations of vehicle sharing systems

We establish asymptotic optimality of base-stock repositioning policy and prove near optimal regret bound of learning

Learning and optimizing in high dimension with censored data is particularly challenging

Summary

Efficient inventory monitoring is critical for successful operations of vehicle sharing systems

We establish asymptotic optimality of base-stock repositioning policy and prove near optimal regret bound of learning

Learning and optimizing in high dimension with censored data is particularly challenging

Summary

Efficient inventory monitoring is critical for successful operations of vehicle sharing systems

We establish asymptotic optimality of base-stock repositioning policy and prove near optimal regret bound of learning

Learning and optimizing in high dimension with censored data is particularly challenging

More Extensions

- Incorporating other controls such as pricing and special incentive programs
- More practical challenges in inventory monitoring
  - Seasonal or non-stationary demand
  - New infrastructure such as charging stations

Thanks for your attention!

Contact: hansheng.jiang@utoronto.ca

Supplementary slides

Numerical Illustration