Learning While Repositioning in On-Demand Vehicle Sharing Networks

Hansheng Jiang
University of Toronto

Joint work with
Chunlin Sun, Max Shen, Shunan Jiang

Vehicle Sharing Networks

Features

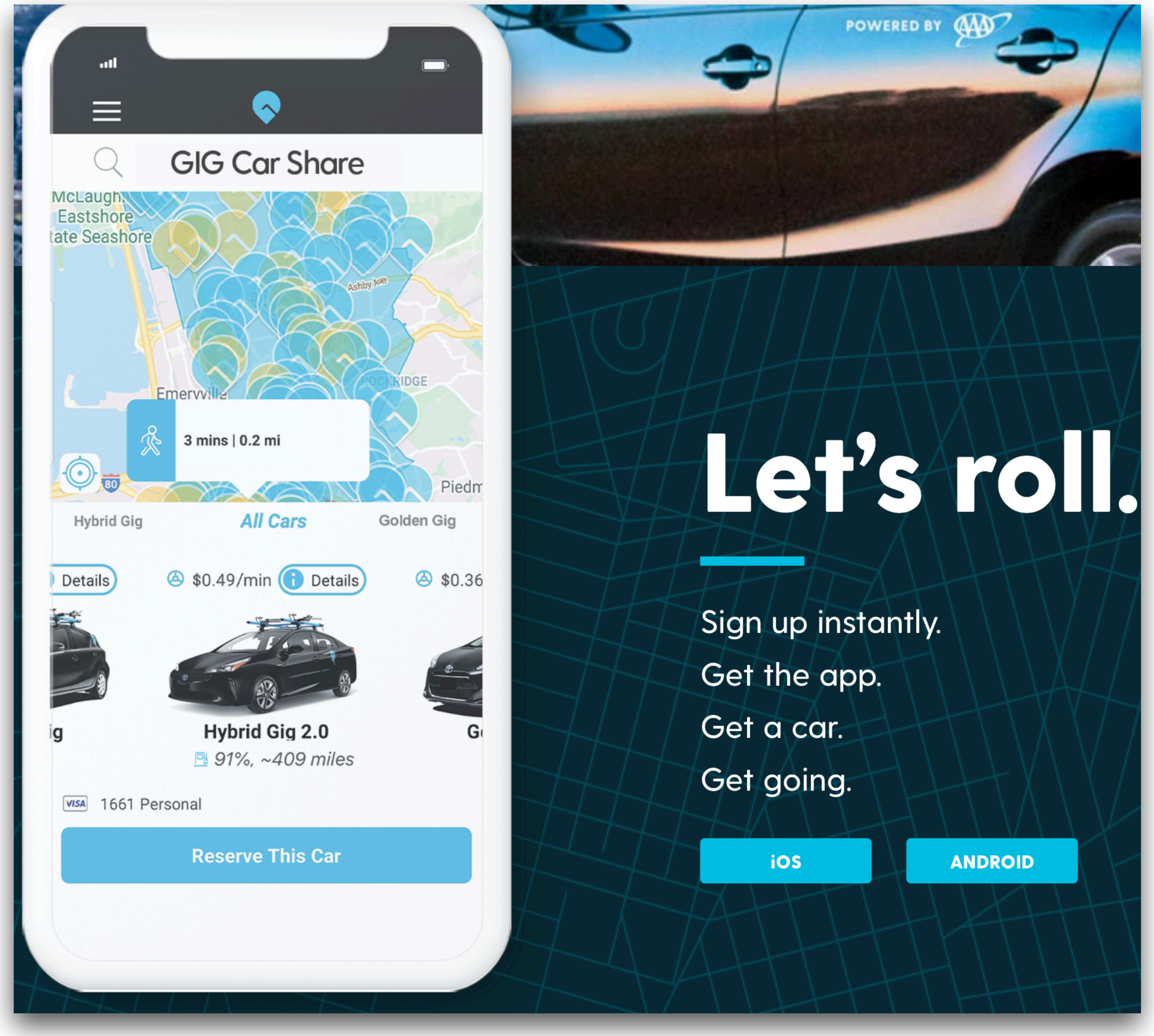
- **On-demand**: customers reserve a vehicle when they want
- **One-way**: rent from one location and return the vehicle to *any other* location in the service network
- Examples: bikes, scooters, cars, and emerging applications of autonomous vehicles

Multifaceted Benefits

- Increased **flexibility** and competitive **costs** for customers
- **Sustainability** benefits
  - May reduce overall vehicle ownerships and produce less carbon emissions
  - Help to promote adoption of electric vehicles equipped with cleaner energy

Source: Generated by Midjourney

Emerging Platforms



Source: gigcarshare.com

Source: evo.ca

GIG Car Share (launched in 2017) is a carsharing service in the San Francisco Bay Area, Sacramento, and Seattle, created by the AAA.

The company operates a fleet of Toyota Prius **Hybrid vehicles** and **all-electric** Chevrolet Bolts.

Evo Car Share (launched in 2015) is a carsharing service in Greater Vancouver and Victoria, created by the BCAA.

The company offers exclusively Toyota Prius **Hybrid** vehicles.

Emerging But No Success?

On July 25, GIG Car Share announces shut-down by end of 2024…

it was much cheaper than ride share for most medium length trips, sad to see it go

242 Reply Award Share ···

Competitive Prices

wait what???? this was so helpful to me i don't wanna go back to paying for lyfts everywhere

47 Reply Award Share ···

This really sucks. As a non car owner and, with ride shares being crazy expensive in this city, this was my real only quick, cheap-ish option to get from A to B. With this, Car2Go, and ReachNow all bailing you have to wonder if we'll see another car share company pop up.

156 Reply Award Share ···

Reduce Car Ownerships

Really disappointed to get this news -- these cars have been a lifesaver for me over the past couple of years. Wonder what they are going to do with all those Priuses.

100 Reply Award Share ···

huge bummer 😭😭😭, gig cars are such a convenient option for one-way trips.

i don't know what their numbers are like, but it sounds like a good amount of their usage was people commuting to and from work.

22 Reply Award Share ···

Great for Commute

These are great for last mile trips in Oakland/Berkeley where public transit is bad or the bike infrastructure isn't great (which is many places). Hopefully someone else takes up the niche in the market.

11 Reply Award Share ···

Last-Mile Trips

Yet, GIG states that…

> *Despite our best efforts, there have been challenges — primarily around decreased demand, rising operational costs, and changes to consumer commuting patterns.*

Operational Challenges

Numerous **operational challenges** of vehicle sharing networks

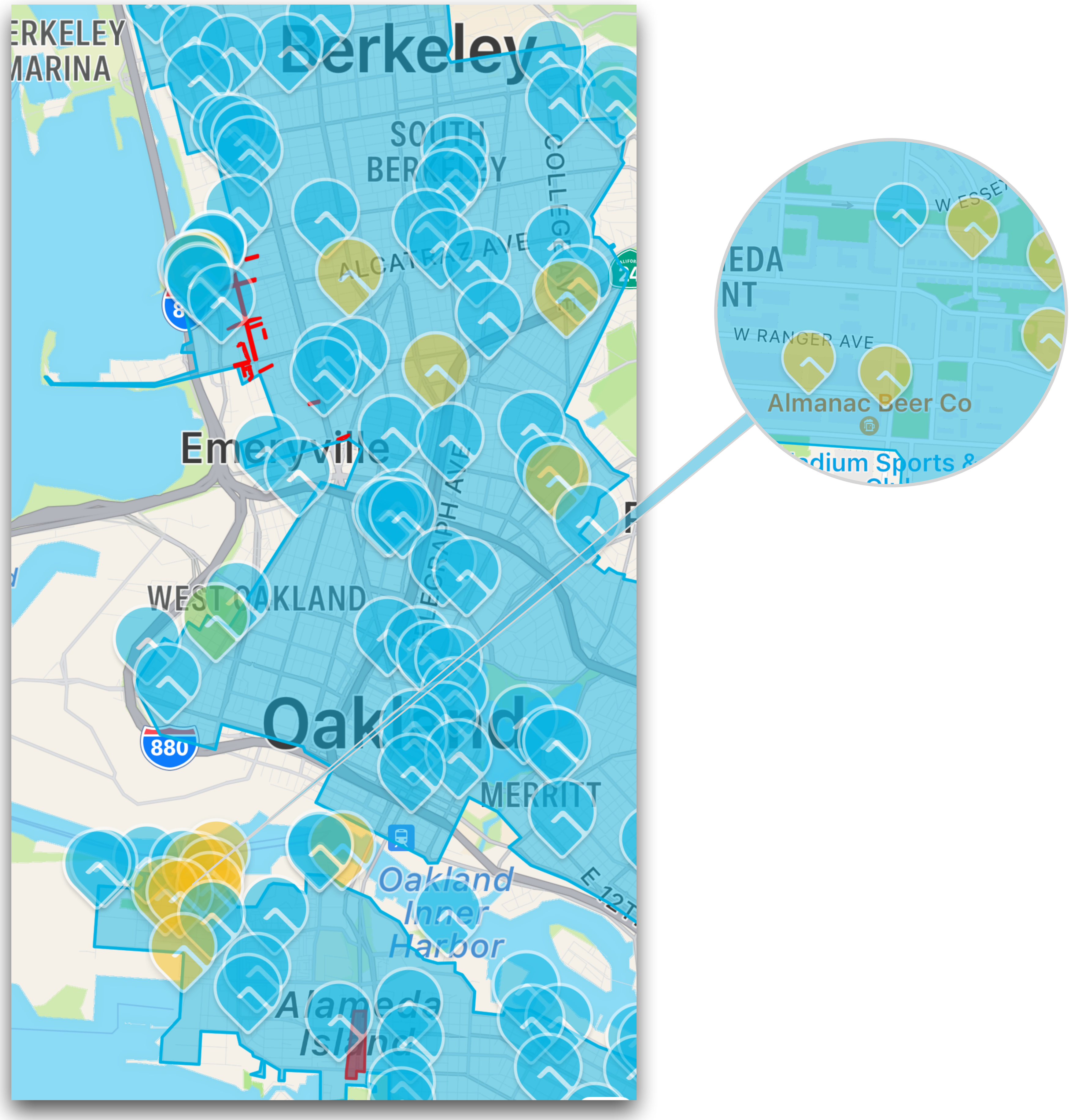
- Service region design
- Fleet sizing / staffing
- Trip pricing (fixed / dynamic / subscription)
- Infrastructure planning, e.g., battery / charging station

……

Focus of this talk: Inventory Repositioning

Why 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

Anecdotal example of low utilization: oversupply of vehicles near brewery

Matching Supply with Demand in Network

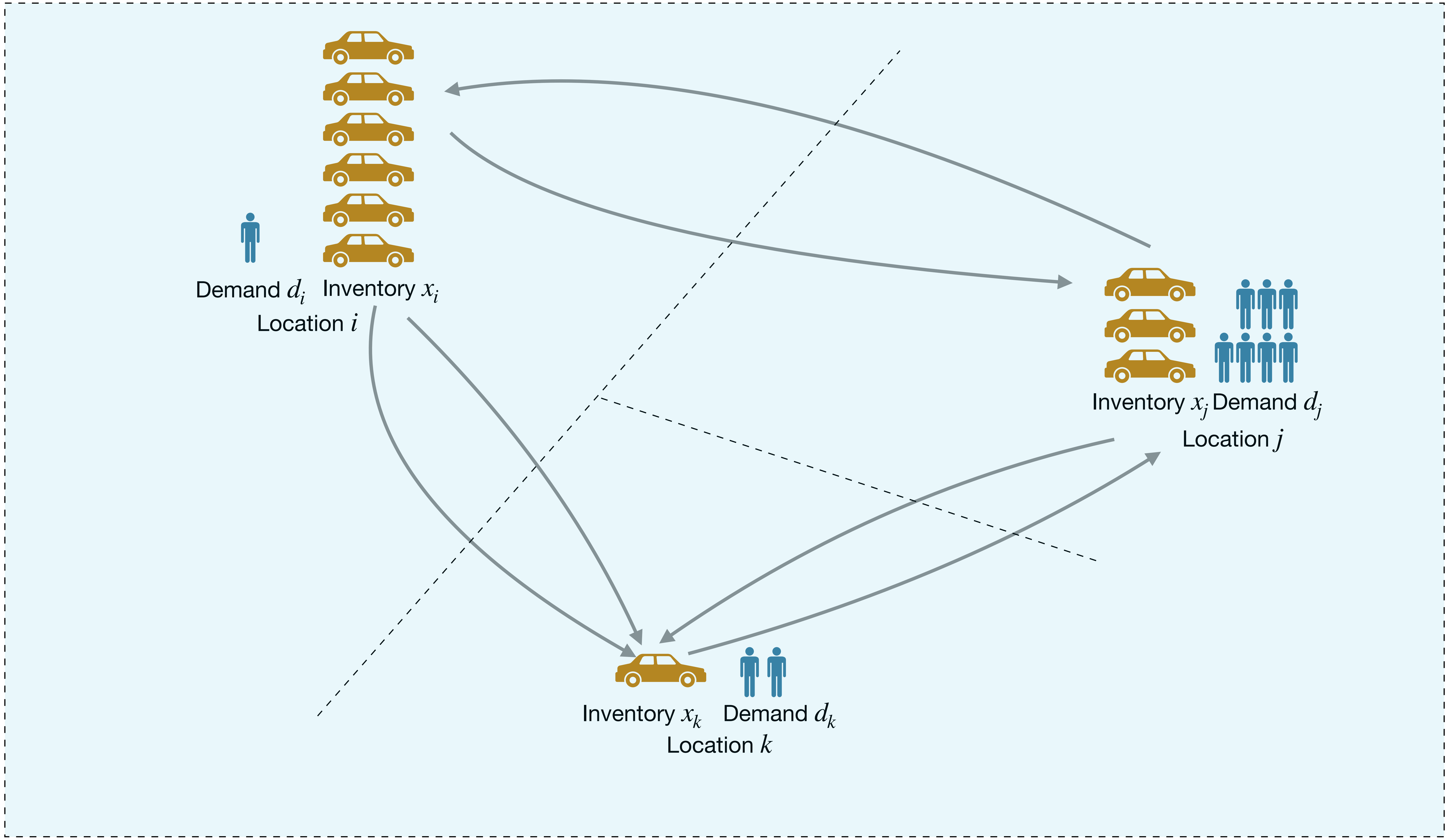


Illustration of 3 locations in a n-location service region

Vehicle Repositioning from the Lens of Inventory Control

Network Inventory Dynamics as Markov Decision Process

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

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

$$\Delta_{n-1} = \{(z_1, \ldots, z_n) \mid \textstyle\sum_i z_i = 1, z_i \geq 0\} \quad \text{(State Space)}$$

II) Service provider makes a decision on the target repositioning inventory level $\boldsymbol{y}_t$ (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 $\boldsymbol{x}_{t+1}$

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

Origin-to-destination matrix for vehicles returning $\boldsymbol{P}_t$

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

Objective

Single-period cost of policy π

Total cost C_t^π = Repositioning cost $M_t(\boldsymbol{y}_t^\pi - \boldsymbol{x}_t^\pi)$ + Lost sales cost $L_t(\boldsymbol{y}_t^\pi)$

Given target repositioning level, complete repositioning by solving minimum cost flow

$$M_t(\boldsymbol{y}_t^\pi - \boldsymbol{x}_t^\pi) = \min \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} \cdot \xi_{ij}$$

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

$$\xi_{ij} \geq 0$$

Censored demand realized and lost sales cost incurred

$$L_t = \sum_i \sum_j l_{ij} \cdot P_{ij}(d_{t,i} - y_{t,i}^\pi)^+$$

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

Optimal Policy (that minimizes long run average cost)

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

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

Base-Stock Repositioning Policy

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

Base-Stock Repositioning Policy

Several Advantages

- Easy to interpret and implement in practice
- State-independent policy
- Rich literature in classic inventory control

What about the performances of Base-Stock Repositioning Policy?

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 I

Theorem (Asymptotic Optimality I, informal)

The ratio of the best base-stock repositioning policy's costs against that of the optimal policy converges to 1 when the lost sales cost becomes large compared to repositioning cost. More specifically, the ratio

$$\frac{\text{Long run average cost of best base-stock repositioning policy}}{\text{Long run average cost of optimal repositioning policy}} = 1 + \Theta\left(\Gamma^{-1}\right),$$

which approaches 1 as $\Gamma := \sum_{i,j} l_{ij} / \sum_{i,j} c_{ij}$ approaches infinity.

Remark

- **Practical relevance**
  - Large l_{ij}: Priority in minimizing user dissatisfication and need for market growth
  - Small c_{ij}: Repositioning can be done in bulk and thus relatively cost-effective
- **Analogous asymptotic optimality result** in single-product single-location inventory control when the ratio of the lost sales cost and the holding cost goes to infinity

Asymptotic Optimality II

Theorem (Asymptotic Optimality II, informal)

The ratio of the best base-stock repositioning policy's costs against that of the optimal policy converges to 1 when the number of locations n becomes large. More specifically,

$$\frac{\text{Long run average cost of best base-stock repositioning policy}}{\text{Long run average cost of optimal repositioning policy}} = 1 + \Theta\left(n^{-\frac{1}{2}}\right),$$

which approaches 1 as n approaches infinity.

Remark

- **Intuition:** Lost sales cost incurred individually at each location — the opposite of "risk pooling"
- **Operational value:** Achieve asymptotic optimality in this analytically-challenging regime with large n

Learning Best Base-Stock Policy on the Fly

Performance Metric

The regret is the difference in costs incurred by algorithm A compared with that of the base-stock repositioning policy with optimal base-stock level $S^\star$

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

Challenges of Learning While Repositioning

- Demand distribution is unknown and censored demand is observed
- Randomness in both demand arriving and vehicle returning
- Network contains multiple locations and limited (fixed) supply

Learning While Repositioning Problem

Base Stock Level $\mathbf{S}_1$ → Base Stock Level $\mathbf{S}_2$ → Base Stock Level $\mathbf{S}_3$ …… Base Stock Level ?

Single-period cost * $C_1^{\mathbf{S}_1}$

Single-period cost $C_2^{\mathbf{S}_2}$

Single-period cost $C_3^{\mathbf{S}_3}$

……

?

Which level to experiment with next?

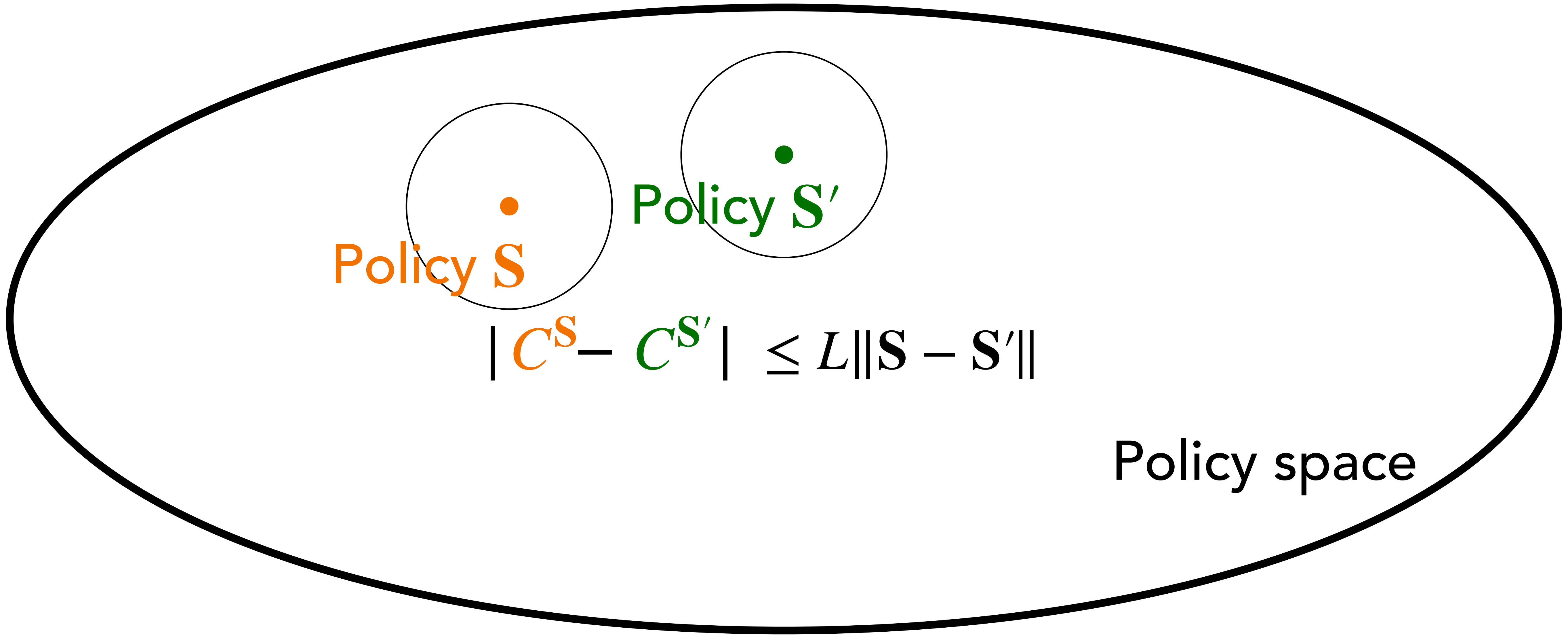
* With only censored demand, lost sales costs in single-period costs are not observable, but we can circumvent this by defining **modified costs** $\tilde{C}_t = C_t - \sum_{i,j} l_{ij} P_{t,ij} d_{t,i}$ that is **observable** and does not affect regret value

First Attempt

A Natural Bandit Learning Perspective

- Treat each base-stock repositioning policy as an arm
- The reward of each arm is negative long-run average cost
- View negative single-period cost as a noisy observation of reward

Lipschitz Bandits-based Repositioning (LipBR) Algorithm



Key Idea

Choose the next policy as the arm guided by Lipschitz bandits framework

Regret Analysis of LipBR

Lipschitz Bandits-based Repositioning (LipBR) Algorithm

1. Establish Lipschitz property of the long-run average cost wrt policy
2. Discretize the policy space Δ_{n-1} by covering, and bound the covering number by $O(\epsilon^{1-n})$ for accuracy ϵ
3. Concentration inequalities of single period costs versus long-run average costs
4. Regret $\approx \sqrt{KT} + K\epsilon$, where $K = O(\epsilon^{1-n})$ and $\epsilon = O(T^{-1/(n+1)})$

LipBR: Regret with Critical Dependence on n

Theorem (Regret of LipBR, informal)

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

Remark

- **Pros:** LipBR is based on a very natural idea of bridging bandits and MDP. It works under the most general network and cost structure
- **Cons:** The regret has a critical dependence on the number of locations n. When n is large, the regret guarantee is almost linear

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

Inherent Complexity of Learning While Repositioning

Proposition (A Negative Example)

There exists a set of two-dimensional joint distribution $\mathscr{P}$ such that for any $(x_0, y_0) \in \{(x_0, y_0) : x_0 + y_0 = 1, x_0, y_0 \geq 0\}$, the censored distribution of $\left(\min(X, x_0), \min(Y, y_0)\right)$ is the same for all $(X, Y) \in \mathscr{P}$.

Remark

Learning **joint demand** distributions with **multi-dimensional** censored demand data but a **limited supply** is inherently impossible.

To reduce regret, we need to introduce additional conditions and employ the problem structure……

But, what kind of condition/structure?

Let's Restart with the Offline Problem

Offline problem solves for $\mathbf{S}^\star$ with uncensored demand

$$\min_{\mathbf{S}} \quad \frac{1}{t}\sum_{s=1}^{t} C_s(\boldsymbol{x}_s, \mathbf{S}, \boldsymbol{d}_s, \mathbf{P}_s)$$

$$\text{s.t.} \quad \boldsymbol{x}_{s+1} = (\mathbf{S} - \boldsymbol{d}_s)^+ + \mathbf{P}_s^T \min(\mathbf{S}, \boldsymbol{d}_s), \text{ for all } s = 1, \ldots, t-1$$

$$\mathbf{S} \in \Delta_{n-1}$$

Even the offline problem with uncensored demand is not trivial!

- The decision variable $\mathbf{S} \in \Delta_{n-1}$ is continuous n-dimensional
- The offline problem is non-convex in $\mathbf{S}$ because of $()^+, \min$

Solving the Offline Problem

Two Reformulations Tackling NonConvexity

MILP Reformulation

- Introduce binary auxiliary variables to express nonconvex piecewise linear functions originated from demand censoring
- $O(n^2t + nt^2)$ constraints
- $O(n^2t)$ decision variables

LP Reformulation

- Under a mild **cost condition***
  $\sum_{i=1}^{n} l_{ji}P_{t,ji} \geq \sum_{i=1}^{n} P_{t,ji}c_{ij}$
- The resulting LP contains $O(nt)$ constraints and $O(n+t)$ decision variables, and can be solved efficiently

Generalization Bound of Offline Solution

- We prove the offline solution enjoys a tight generalization bound $O(\sqrt{\log T}/\sqrt{t})$ with probability at least $1 - T^{-2}$.

---

* This cost condition holds easily in practice and aligns well with one regime that base-stock policy is optimal. Similar conditions have been used in the literature as well.

Two Algorithms Based on Offline Solution

If demand is uncensored…

Dynamic Learning Algorithm

1. Employ doubling epoch scheme so that new policy can dominate the regret rate
2. At beginning of each epoch, solve the offline problem and apply the updated policy in the whole epoch

If demand is censored but network independence holds…

One Time Learning Algorithm

1. Explore for $nT^{2/3}$ time periods by placing sufficient inventory in n locations respectively to construct $T^{2/3}$ effective uncensored network demand
2. Solve the offline problem using constructed data
3. Exploit the policy learned from the offline problem in remaining periods

Regret Analysis of Dynamic Learning and One-Time Learning

- By proving a tight generalization bound of offline solution, we can derive the following regret guarantees

Theorem (Regret of Dynamic Learning, informal)

Under the oracle of uncensored demand data, the dynamic learning algorithm can achieve $\tilde{O}(T^{\frac{1}{2}})$ regret.

Theorem (Regret of One-Time Learning, informal)

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

Remark

- Learning while repositioning is easy in the oracle of uncensored demand
- The one-time learning algorithm requires $O(T^{2/3})$ periods to collect data location by location and thus incurs a suboptimal regret compared to the oracle

Comparison of Two Algorithms

	Dynamic Learning Algorithm	One-Time Learning Algorithm
Assumption	Requires uncensored demand	Requires network independence
Data Access	Oracle of uncensored demand	Network independence allows pure exploration to collect uncensored data
Policy Update	Compute offline solution again each time with new uncensored data	Compute offline solution once
Regret	$\tilde{O}(T^{\frac{1}{2}})$	$\tilde{O}(T^{\frac{2}{3}})$

Going Beyond Offline Solution

Can we design an algorithm that achieves regret guarantee of $O(T^{\frac{1}{2}})$ **without** both uncensored demand and network independence?

Yes!

(Under the same mild cost condition used in LP reformulation)

Online Gradient Repositioning (OGR)

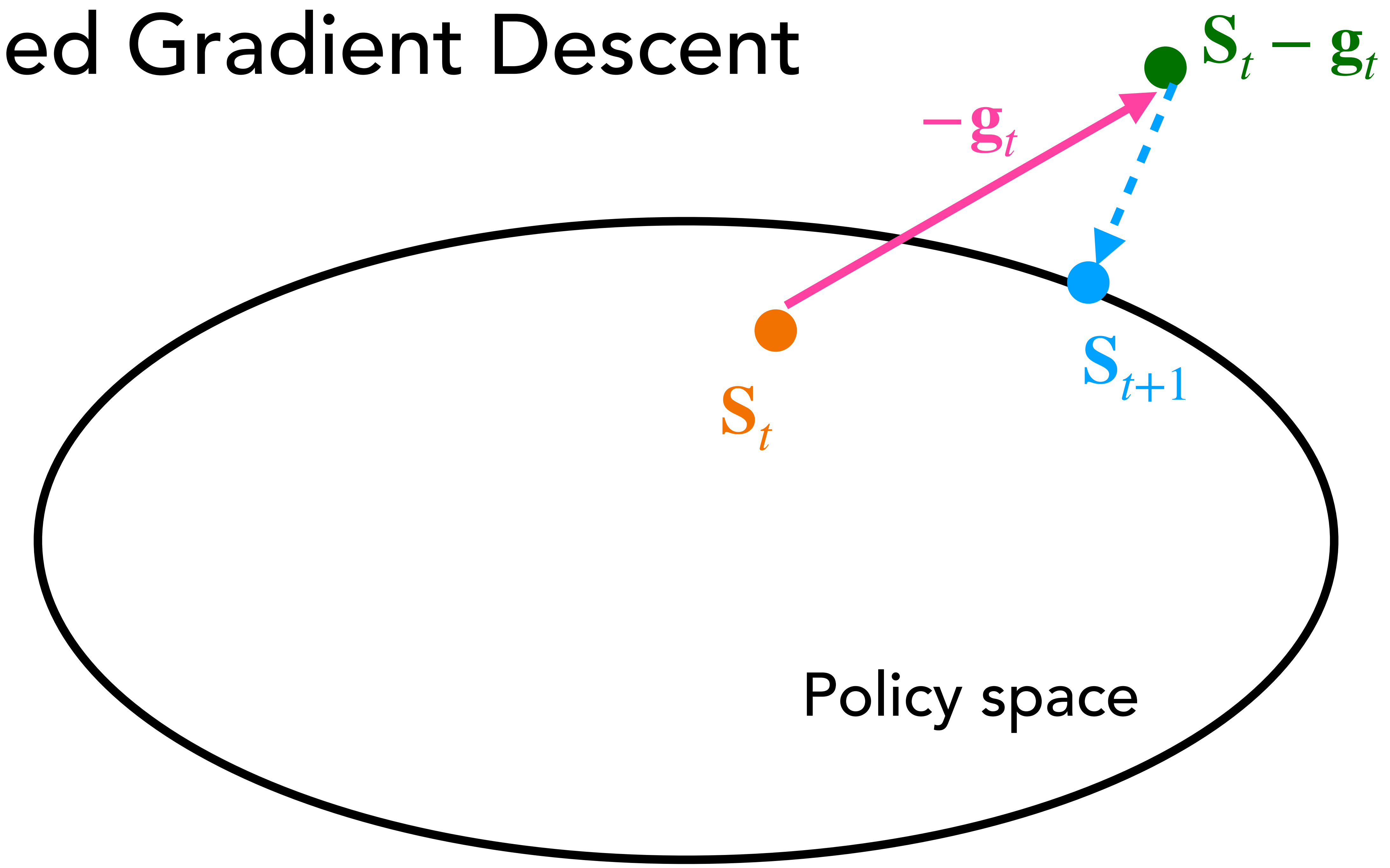
Theorem (Regret of OGR, informal)

Our Online Gradient Repositioning **(OGR)** algorithm achieves a regret of $O(T^{\frac{1}{2}})$ and this rate even holds for adversarial data.

This rate **matches** the theoretical lower bound.

Algorithm Design of OGR

Framework: Projected Gradient Descent



Key Challenges Addressed

- How to define the gradient, with only censored demand?
- How to disentangle intertemporal dependence in regret analysis?

Algorithm Design of OGR

At iteration t

1. Compute the dual optimal solution $\lambda_{t,i}$ to the constraints $w_{t,i} \leq \min\{d_{t,i}, S_i\}$ in a small linear program
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 $\mathbf{S}_{t+1}$

Significant Advantages of OGR

Best of Many Worlds

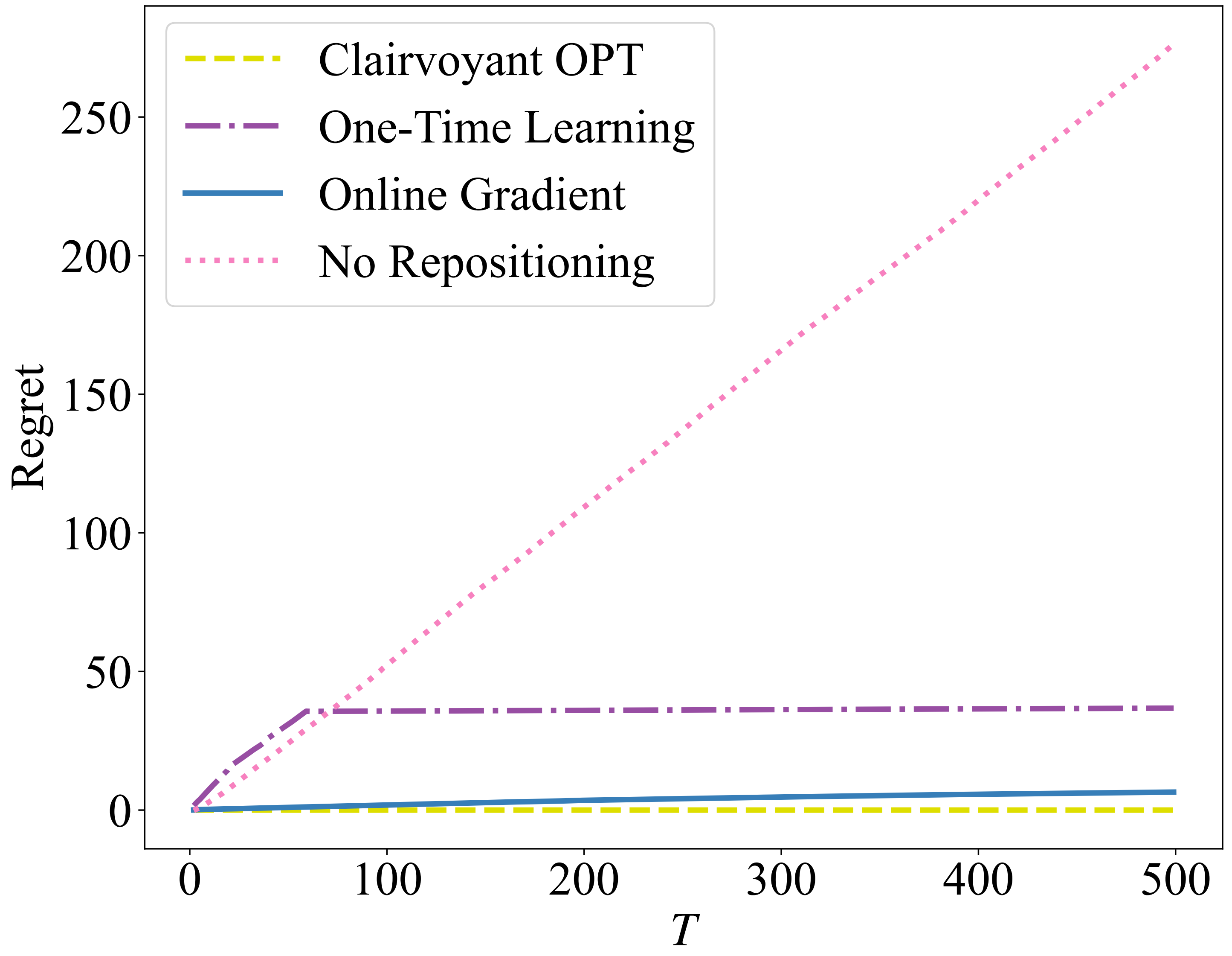
- **Minimal Data Requirement**
  - Utilizing only censored demand data
- **Computational Efficiency**
  - In each period, only computes one small linear program with $O(n^2)$ constraints and variables, which *does not scale up* with time horizon
- **Reliability**
  - Regret guarantee for both i.i.d. and non-i.i.d. (adversarial) demands and transition probabilities

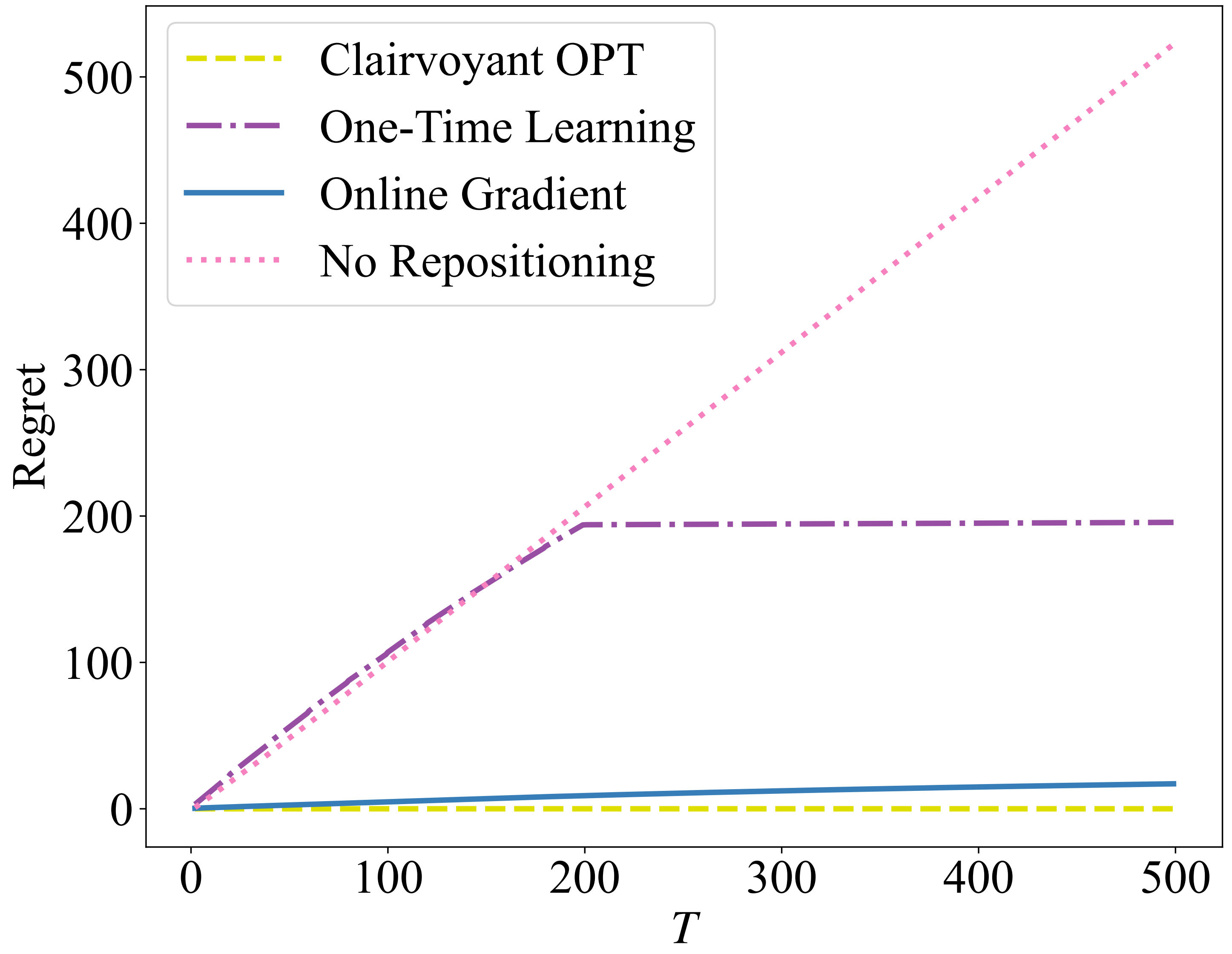
Numerical Illustration

$n = 3$

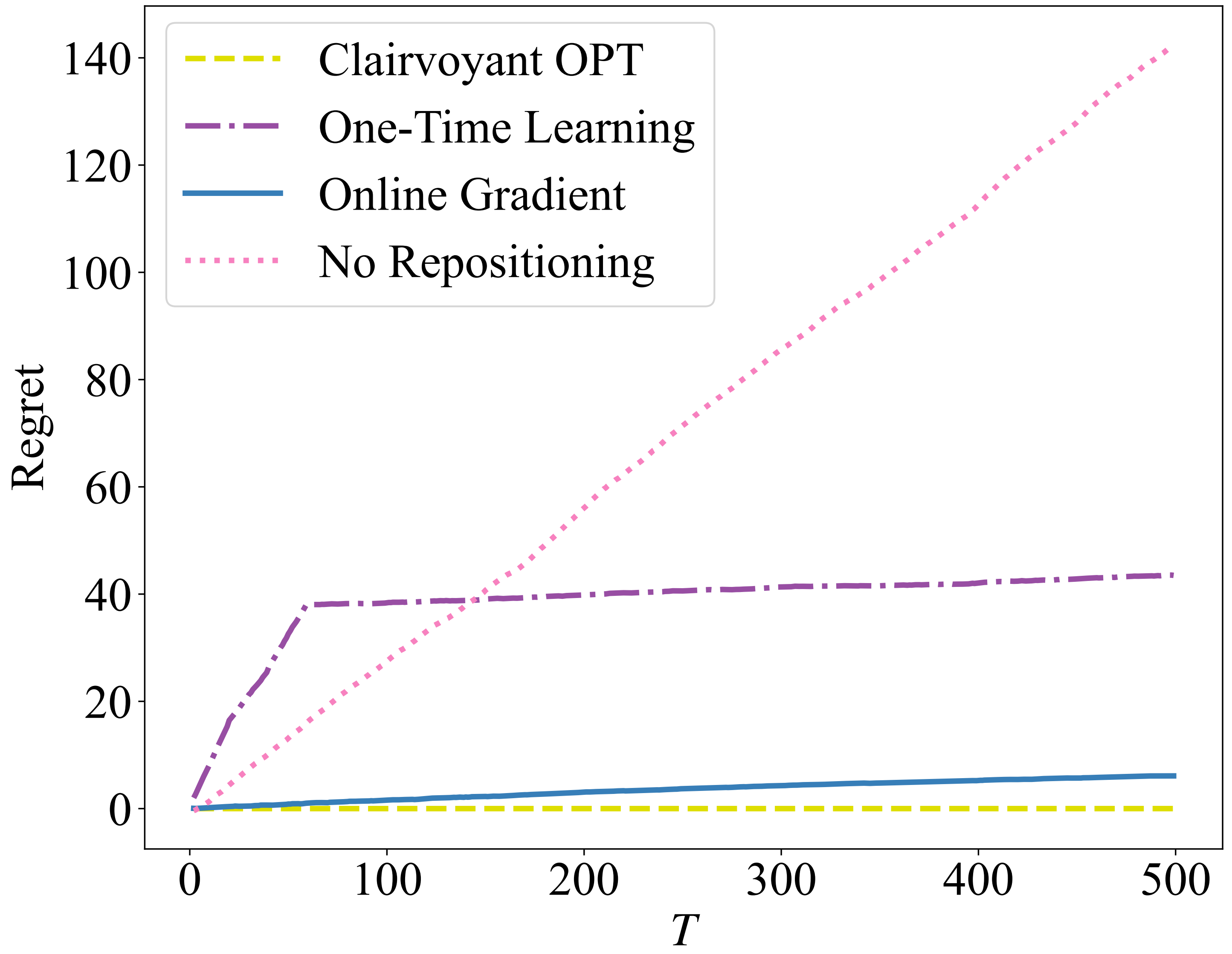
$n = 10$

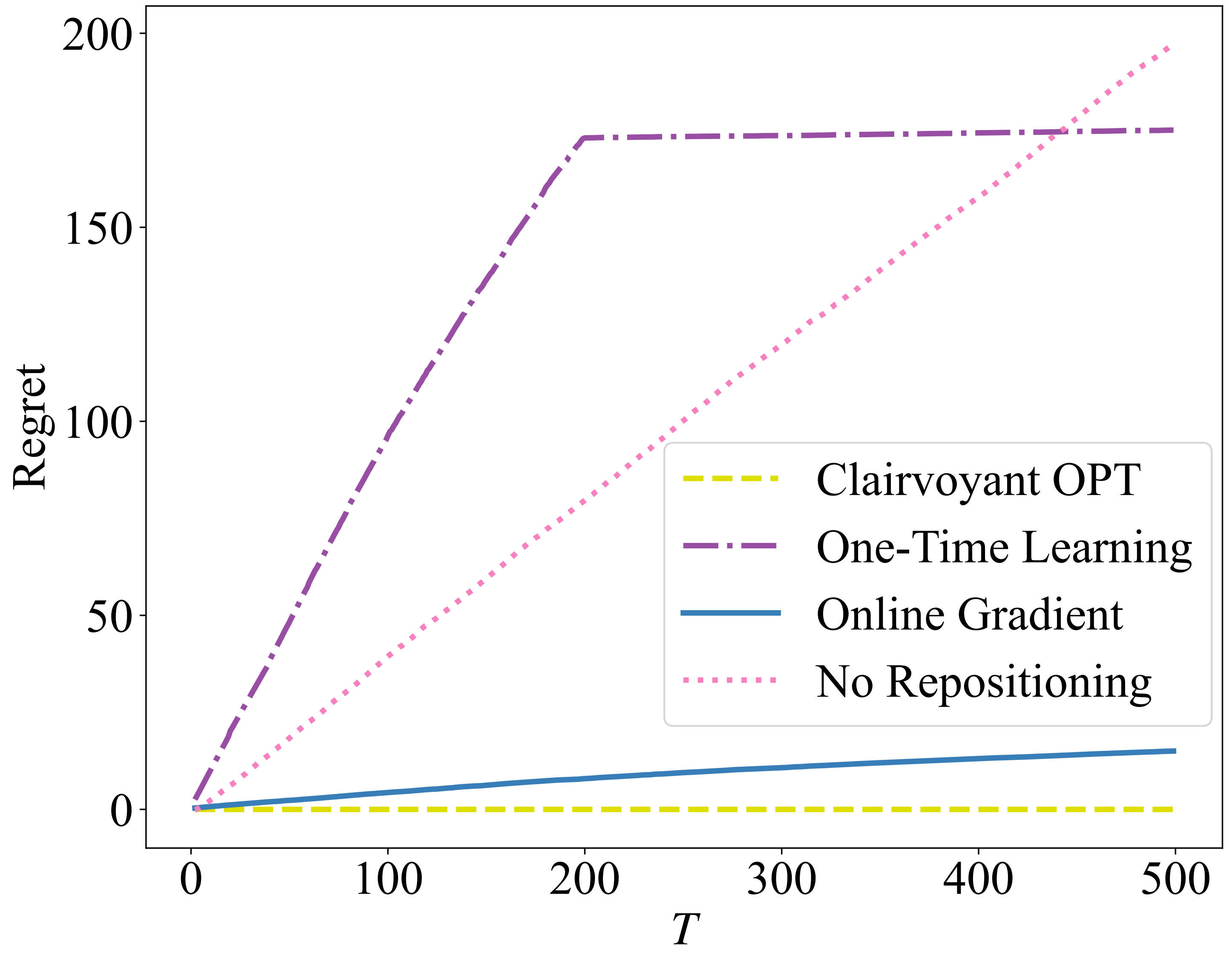
Under network independence





Without network independence





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

Takeaway

For practitioners, our analysis indicates that it is generally challenging to match supply and demand in a vehicle sharing network, especially given that the supply is constrained

Our results urge more powerful data analytic tools to reduce operational costs and improve system efficiency in vehicle sharing

Thanks for your attention!

Contact: hansheng.jiang@utoronto.ca

Supplementary slides

Details of LP Reformulation

Suppose $\sum_{i=1}^{n} l_{ij} P_{s,ij} \geq \sum_{i=1}^{n} P_{s,ij} c_{ji}$ the offline problem can be reformulated into LP

$$\min \sum_{s=1}^{t} \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} \xi_{s,ij} - \sum_{s=1}^{t} \sum_{i=1}^{n} \sum_{j=1}^{n} l_{ij} P_{s,ij} w_{s,i}$$

$$\text{subject to } \sum_{i=1}^{n} \xi_{s,ij} - \sum_{k=1}^{n} \xi_{s,jk} = w_{s,j} - \sum_{i=1}^{n} P_{s,ij} w_{s,i}, \text{ for all } j = 1,\ldots,n \text{ and } s = 1,\ldots,t,$$

$$\xi_{s,ij} \geq 0, \forall i = 1,\ldots,n, \text{for all } i,j = 1,\ldots,n \text{ and } s = 1,\ldots,t,$$

$$\sum_{i=1}^{n} S_i = 1, \ \{S_i\}_{i=1}^{n} \in [0,1]^n,$$

$$w_{s,i} \leq \min\{d_{s,i}, S_i\}, \ w_{s,i} \geq 0, \ \text{ for all } s = 1,\ldots,t, i = 1,\ldots,n\,.$$

Details of MILP Reformulation

$$\min \sum_{s=1}^{t}\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}\xi_{s,ij} - \sum_{s=1}^{t}\sum_{i=1}^{n}\sum_{j=1}^{n} l_{ij}P_{s,ij}m_{s,i} + \sum_{s=1}^{t}\sum_{i=1}^{n}\sum_{j=1}^{n} l_{ij}P_{s,ij}d_{s,i}$$

subject to $\sum_{i=1}^{n}\xi_{s,ij} - \sum_{k=1}^{n}\xi_{s,jk} = m_{s,j} - \sum_{i=1}^{n}P_{s,ij}m_{s,i},$ for all $j = 1,\ldots,n$ and $s = 1,\ldots,t,$

$\xi_{s,ij} \geq 0, \forall i = 1,\ldots,n,$ for all $j = 1,\ldots,n$ and $s = 1,\ldots,t,$

$\sum_{i=1}^{n} S_i = 1, \mathbf{S} = \{S_i\}_{i=1}^{n} \in [0,1]^n,$

$(m_{1,i}, m_{2,i}, \ldots, m_{t,i})^T = \mathbf{\Gamma}_i^T(\tilde{m}_{1,i}, \tilde{m}_{2,i}, \ldots, \tilde{m}_{t,i})^T$ for all $i = 1,..,n,$

$\mathbf{\Gamma}_i(d_{1,i}, d_{2,i}, \ldots, d_{t,i})^T = (\tilde{d}_{1,i}, \tilde{d}_{2,i}, \ldots, \tilde{d}_{t,i})^T$ for all $i = 1,..,n,$

$\sum_{s=1}^{t} z_{s+1,i}\cdot\tilde{d}_{s,i} \leq S_i \leq \sum_{s=1}^{t} z_{s,i}\cdot\tilde{d}_{s,i} + z_{t+1,i},$ for all $i = 1,\ldots,n,$

$-2(1 - z_{s',i}) \leq \tilde{m}_{s,i} - S_i \leq 2(1 - z_{s',i}),$ for all $1 \leq s' \leq s \leq t$ and $i = 1,..,n,$

$-2(1 - z_{s',i}) \leq \tilde{m}_{s,i} - \tilde{d}_{s,i} \leq 2(1 - z_{s',i}),$ for all $1 \leq s < s' \leq t+1$ and $i = 1,..,n,$

$\sum_{s=1}^{t+1} z_{s,i} = 1,$ for all $i = 1,\ldots,n,$

$\mathbf{z}_s = \{z_{s,i}\}_{i=1}^{n} \in \{0,1\}^n,$ for all $s = 1,\ldots,t+1.$

- Note: $\mathbf{\Gamma}_i$ is permutation matrix

Generalization Bound

Theorem (Generalization Bound, informal)

With probability at least $1 - \frac{1}{T^2}$, it holds for all $\mathbf{S}$ simultaneously that

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

OGR Algorithm Details

Algorithm 1 `OGR`: Online Gradient Repositioning Algorithm

Input: Number of iterations T, initial repositioning policy $\boldsymbol{y}_1$;

for $t = 1, \ldots, T$ **do**

  Set the target inventory be $\boldsymbol{y}_t$ and observe realized censored demand $\boldsymbol{d}_t^c = \min(\boldsymbol{y}_t, \boldsymbol{d}_t)$;

  Denote $\boldsymbol{\lambda}_t = (\lambda_{t,1}, \ldots, \lambda_{t,n})^\top$ be the optimal dual solution corresponding to constraints (29)

$$\widetilde{C}_t(\boldsymbol{x}_{t+1}, \boldsymbol{y}_t, \boldsymbol{d}_t, \boldsymbol{P}_t) = \min \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} \xi_{t,ij} - \sum_{i=1}^{n} \sum_{j=1}^{n} l_{ij} P_{t,ij} w_{t,i} \tag{28}$$

$$\text{subject to} \sum_{i=1}^{n} \xi_{t,ij} - \sum_{k=1}^{n} \xi_{t,jk} = w_{t,j} - \sum_{i=1}^{n} P_{t,ij} w_{t,i}, \ \text{for all } j = 1, \ldots, n,$$

$$w_{t,i} \geq 0, \ \xi_{t,ij} \geq 0, \ \text{for all } i, j = 1, \ldots, n,$$

$$w_{t,i} \leq (\boldsymbol{d}_t^c)_i, \quad \text{for all } i = 1, \ldots, n, \tag{29}$$

  where $\boldsymbol{\xi_t} = \{\xi_{t,ij}\}_{i,j=1}^n, \boldsymbol{w}_t = \{w_{t,i}\}_{i=1}^n$ are decision variables;

  Compute the gradient $\boldsymbol{g}_t = (g_{t,1}, \ldots, g_{t,n})^\top$, where $g_{t,i} = \lambda_{t,i} \cdot \mathbb{1}_{\{(\boldsymbol{d}_t^c)_i = y_{t,i}\}}$, for all $i = 1, \ldots, n$;

  Update the repositioning policy $\boldsymbol{y}_{t+1} = \Pi_{\Delta_{n-1}} \left(\boldsymbol{y}_t - \frac{1}{\sqrt{t}} \boldsymbol{g} \right)$;

end for

Output: $\{\boldsymbol{y}_t\}_{t=1}^T$.

Steps of Dynamic Learning and One-Time Algorithm

If demand is uncensored…

Dynamic Learning Algorithm

1. Employ doubling epoch scheme so that new policy can dominate the regret rate
2. At beginning of each epoch, solve the offline problem and apply the updated policy in the whole epoch

If demand is censored but network independence holds…

One Time Learning Algorithm

1. Explore for $nT^{2/3}$ time periods by placing sufficient inventory in n locations respectively to construct $T^{2/3}$ effective uncensored network demand
2. Solve the offline problem using constructed data
3. Exploit the policy learned from the offline problem in remaining periods