Improving The Repositioning Operation For Bike Rental System

Kevin Huang & Zitao Shen

May 2019

Abstract

Recent years, dockless and dock based bike-rental systems are widely deployed in major cities around the world. However, due to a potential mismatch of rental/return demand versus bike/dock availability in the current bike rental system, operating an inventory rebalancing strategy could be critical for maintaining a large-scale service network. Therefore, this paper served as an initial step on designing some implementable bike rebalancing policies. By analyzing the impact of capacity constraints on the repositioning policy under two regions setting, we proposed a two-threshold policy for dock-based system, which is similar to the existing two-threshold policy for dockless system in the literature, to control the number of bikes in each region in a particular range where there's neither too many nor too few bikes in each station (known as no-repositioning region). In the future, this project could potentially be served as a better motivating example in such product-rental system with capacity constraints, such as the charging allocation for electric vehicles.

1 Introduction

Nowadays, bike-rental systems are widely deployed in many major cities around the world. The reason for the high popularity of this idea is that there are many benefits associated with the bike rental idea. For example, a bike-rental system can have positive effects on public health by increasing daily access to bikes. Also, to compare with other transit, bikes are commonly considered to be environmental-friendly by the public. The booming demand accelerates the development of the bike-rental system. In the early stage, the bike-rental system is composed of several bike stations with limited docks. From bike-rental companies' side, although it is easy for them to manage those bikes, to users, their demand on docks, i.e., the "return demand," may not be satisfied when docks of return stations are all occupied. Also, companies may miss a lot of demands from areas, which are far from the existed bike stations.

In the past few years, some new bike-rental companies, like Ofo, begin to offer dockless bikes. In their systems, it does not have any stations or docks.

Therefore, people can have more freedom in returning their bikes. However, the trade-off is that since bikes' locations are not fixed, it increases the cost for the user to find the bike, and for the companies to manage the inventory among all possible sites. Existing popular bike-rental systems operating either dock-based or dockless have their pros and cons. Therefore, some companies give out their solutions. For example, in Minneapolis, the bike-sharing company, Nice Ride, offers a mixed biking sharing service. Besides the mature dock based bike sharing system, they also provide dockless bikes, and they ask their users to return their bikes to some specific areas; otherwise, there will be some penalties added to users' account.

Based on the discussion above, several interesting questions are raised. First, since there is a potential mismatch of rental/return demand versus availability in both systems, operating an inventory rebalancing strategy could be beneficial. So how will their optimal policy differ in nature? Will one outperform the other operating such policy in terms of cost? In our proposed project, we hope to address some of those questions, or at least offer some insights on designing bike inventory rebalancing strategies. Due to the short time for the project, we wish to narrow our focus on more straightforward model settings. For example, to study the difference in the inventory rebalancing policy, we will concentrate on one of the significant differences between the two systems: whether the system is constrained with return capacity limit and incurred a "lost return" cost. The technical approach can be described as follows: first, we refine an existing inventory rebalancing strategy between two locations by adding capacity constraints to fit our model for both bike rental systems. Then, for empirical study, we will utilize the rental activity record from Minnesota's bike rental company Nice Ride to study the pattern of the real demand. Later, this project uses that information to generate necessary parameters in the simulation under the asymmetric demand assumption. In the next section, the paper presents an overview of the related literature. Section 3 formulates the problem as an MDP problem. In Section 4, we carry out an extensive simulation study based on the Nice Ride's data. Finally, Section 5, the paper summarizes and conclude the results of the simulation study.

2 Literature Review

In the recent year, due to the appearance of the sharing economy, a lot of innovation bursts in the shared mobility system, such as car sharing, ride sharing and so on. As one of the critical issues, the inventory rebalancing repositioning has naturally gathered a lot of attention from the operation management community. For this project, there are three papers mainly considered. In the context of the theoretical framework for inventory rebalancing, Benjaafar et al. (2018) [1] and He et al. (2018) [2] both study the structural properties of the optimal repositioning car policy under a car rental network setting by formulating as a dynamic program model. In terms of assumptions, they both assume that lost sales cost outweighs the repositioning cost so rental companies can have the motivation to reposition their cars.

On the other hand, to compare with He et al. (2018), by allowing the rental period to be greater than one, Benjaafar et al. (2018) provides a more general characterization, since they focus more on finding theoretical characterization of the optimal policy. In return, Benjaafar et al. (2018) comes up with a more complex formulation for the problem than He et al. (2018), and utilize approximate dynamic programming approaches to approximate their convex objective function. However, He et al. (2018) emphasizes on a computationally efficient solution approach for a 2-region case and multiple region case. Specifically, the 2-region case and the multiple regions cases are formulated as an MDP and a distributionally robust optimization problem respectively. For our project, since bikes' rental periods are usually shorter than cars, and we want our model to be relatively simple, we adopt the stochastic dynamic programming model mentioned in He et al. (2018) as a 2-region dockless system, and add capacity constraints of each location to model the dock based system.

For the empirical study, it is essential to understand how bike rental companies, like Nice Ride, rebalance their bikes under real-life settings. Wang et al. (2018) [3] provides systematic analysis on bike related data and propose BRAVO – the first practical data-driven bike rebalancing app for operators to reposition bikes optimally in real time. The BRAVO evaluates bike rental record from rental companies, including Nice Ride, to set up a rebalancing interval for each station. Then, the app can calculate the optimal rebalancing route between stations by formulating this problem as a Travelling Salesman Problem, where each path's weight is the Euclidean distance between repositioning track and stations. Next, Bravo will set rebalancing number problem as an optimization problem, which the objective is reducing the total amount of loading/unloading bikes. Overall, from this paper, some mentioned empirical policies offer a few insights for the later simulation.

3 Model

3.1 Model Settings

We model the 2-Region repositioning problem as a MDP problem with planning horizon of T periods. Also, there are total N bikes and 2 stations with dock capacity C_1 and C_2 in the system, and the total number of docks (dock capacity C) is fixed throughout all periods. During each period t, d_{1t}, d_{2t} represent the random rental demand in region 1 and 2. $\boldsymbol{d_t}$ is the vector of demands in all regions in period t. Furthermore, $\boldsymbol{d_{[t]}} = (\boldsymbol{d_1, ..., d_t})$ is the historical demand realization up to period t. The demand at both regions during the whole planing horizon $\boldsymbol{d_{[T]}}$ is assumed to follow a known joint probability distribution $\mathbb{P}$. At the start of period t, the company observes the number of bikes in region 1 x_t as system state, then $N - x_t$ is the number of bikes in region 2.

Before the bike demand shows up, the company makes repositioning decision r_t, where r_t bikes are repositioned from region 1 to 2 when $r_t \geq 0$ at cost

$s_{12t} > 0$ per trip per bike, and $r_t \leq 0$ represents repositioning in other direction at cost $s_{21t} > 0$ per trip per bike. Furthermore, based on the fact that the bike company collects rental records, we assume the company has the knowledge over α_{ijt}, which is the probability that a trip begin from region i at period t that ends in region j (in the two region case, the bikes are either returned to its original location or the other location). We also assume all trips should end in one period, so $\sum_j \alpha_{ijt} = 1$. Next, by letting w_{it} be the total fulfilled customer trips from i, $\alpha_{ijt} * w_{it}$ can be the fulfilled customer trips from i to j at period t. For costs related to the mismatch of rental/return demand versus availability in bike system, we defined the lost sale cost and holding cost respectively in the model. In terms of lost sale cost, if a customer, whose intended destination is j, finds no bike available in region i, then a penalty $p_{ijt} > 0$ will incur. To combine with α_{ijt}, we then define $\overline{p}_{it}$,the average lost sale cost of a customer trip originates from i as following:

$$\overline{p}_{1t} = p_{11t}\alpha_{11t} + p_{12t}\alpha_{12t}$$
$$\overline{p}_{2t} = p_{22t}\alpha_{22t} + p_{21t}\alpha_{21t}$$

By adding holding cost h_{it}, as the penalty for arrival costumers finding no dock available for returning bikes at i, we can distinguish the dock based and dockless system. For the dockless system, since costumer can return bikes wherever they what ideally, such cost will not exist. We formulate the following MDP to minimize the expected total cost:

$$V_t(x_t) = \min_{x_t - N \leq r_t \leq x_t} \{\underbrace{s_{12t}r_t^+ + s_{21t}r_t^-}_{\text{Repositioning cost}} + \mathbb{E}_{\mathbb{P}}[J_t(y_t, \boldsymbol{d}_t)]\}$$

where

$$J_t(y_t, \boldsymbol{d}_t) = \min_{w_{1t}, w_{2t}} \{\underbrace{\overline{p}_{1t}(d_{1t} - w_{1t}) + \overline{p}_{2t}(d_{2t} - w_{2t})}_{\text{The lost sale cost}} +$$
$$(\underbrace{h_{1t}(x_{t+1} - C_1)^+ + h_{2t}(N - x_{t+1} - C_2)^+}_{\text{The holding cost}}) * \mathbf{1}_{\text{dock based}} + V_{t+1}(x_{t+1})\}$$

$$\begin{aligned} \text{s.t.} \qquad y_t &= x_t - r_t, \\ x_{t+1} &= y_t - \alpha_{12t}w_{1t} + \alpha_{21t}w_{2t}, \\ w_{1t} &\leq \min(y_t, d_{1t}) \\ w_{2t} &\leq \min(N - y_t, d_{2t}) \end{aligned}$$

where $a^+ = \max(0, a)$ and $a^- = -\min(0, a)$ with terminal cost $V_{T+1}(x_{T+1}) = 0$.

The constraint $y_t = x_t - r_t$ is the target inventory level in region 1 after repositioning. The number of available bikes x_{t+1} for the next period is the sum of the number of available bikes after repositioning: y_t and the net inflow of bikes between two regions: $\alpha_{12t}w_{1t} + \alpha_{21t}w_{2t}$. The fulfilled demand w_{1t} in region 1 should always be no larger than the demand: d_{1t} and post repositioning inventory level: y_t. Same happens to w_{2t}

3.2 Policy Structure

The optimal policy for 2-Region system can be characterized as follows.

Lemma 1. *Suppose $\overline{p}_{it} \geq \alpha_{ijt}(h_{jt} + s_{ji(t+1)})$ for $i, j \in 1, 2$ and $i \neq j$. For each period t, there exist two thresholds $\underline{x}_t, \overline{x}_t$ such that*

$$\begin{aligned} y_t^*(x_t) &= \underline{x}_t, \quad x_t \in [0, \underline{x}_t) \\ y_t^*(x_t) &= x_t, \quad x_t \in [\underline{x}_t, \overline{x}_t] \\ y_t^*(x_t) &= \overline{x}_t, \quad x_t \in (\overline{x}_t, N]. \end{aligned}$$

where $\underline{x}_t, \overline{x}_t$ is the optimal reposition up-to and down-to level solved by the following convex programs:

$$\begin{aligned} \underline{x}_t &= arg \min_{0 \leq y \leq N} \{s_{21t} y + \mathbb{E}_{\mathbb{P}}[J_t(y, \boldsymbol{d}_t)]\} \\ \overline{x}_t &= arg \min_{0 \leq y \leq N} \{-s_{12t} y + \mathbb{E}_{\mathbb{P}}[J_t(y, \boldsymbol{d}_t)]\} \end{aligned}$$

Proof. The proof is very much similar to the proof of Proposition 1 in He et al. (2018), since the only difference between the two models lies in the difference of $J_t(y_t, \boldsymbol{d}_t)$. In the dock-based model, the term $h_{1t}(x_{t+1} - C_1)^+ + h_{2t}(N - x_{t+1} - C_2)^+$ is added to $J_t(y_t, \boldsymbol{d}_t)$. The Lemma 1 in He et al. (2018) states that the value function $V_t(x_t)$ is convex, and therefore $J_t(y_t, \boldsymbol{d}_t)$ is also convex. Clearly adding convex terms $h_{1t}(x_{t+1} - C_1)^+ + h_{2t}(N - x_{t+1} - C_2)^+$ to it won't change its convexity. Now we know that in the dock-based model $\mathbb{E}_{\mathbb{P}}[J_t(y, \boldsymbol{d}_t)]$ is convex. Directly apply Lemma 4 in He et al. (2018) (in Appendix A.2), our Lemma 1 follows by letting $F(y) = \mathbb{E}_{\mathbb{P}}[J_t(y, \boldsymbol{d}_t)]$. □

Similar to the condition emphasized in Benjaafar et al. (2018) and He et al. (2018), the condition $\overline{p}_{it} \geq \alpha_{ijt}(h_{jt} + s_{ji(t+1)})$ requires that the average profit of satisfying a trip originates from i should be higher than the average repositioning cost on returning the bike to i in the next period and the average holding cost for occupying a dock in j at the current period. This condition is critical to the stationary of the whole system, i.e., $p_{ijt} = p_{ij}, s_{ijt} = s_{ij}, h_{it} = h_i$ and $p_{ij} \geq s_{ji} + h_j$.

Under this condition, it will be optimal for the company to implement some rebalancing policies. Here, the optimal rebalancing policy has two thresholds $\underline{x}_t$ and $\overline{x}_t$. When the inventory level in region 1 is below the threshold $\underline{x}_t$, it is optimal to move bikes from region 2 to region 1 to bring the inventory level back to $\underline{x}_t$; When the inventory level in region 1 is above the threshold $\overline{x}_t$, it is optimal to return bikes from region 1 to region 2 to bring down the inventory level to $\overline{x}_t$. If the inventory level of region 1 is in $[\underline{x}_t, \overline{x_t}]$, no repostioning is required, and we refer to this interval as "no-repositioning region", to be consistent with the literature. Hence, for the dock based system, its optimal inventory repositioning policy should have the similar structure as the optimal policy for dockless one which is proved to be a two-threshold policy in He et al. (2018). However, unlike the dockless one, the lower and upper threshold should be expected to change with dock related variable under the dock based settings.

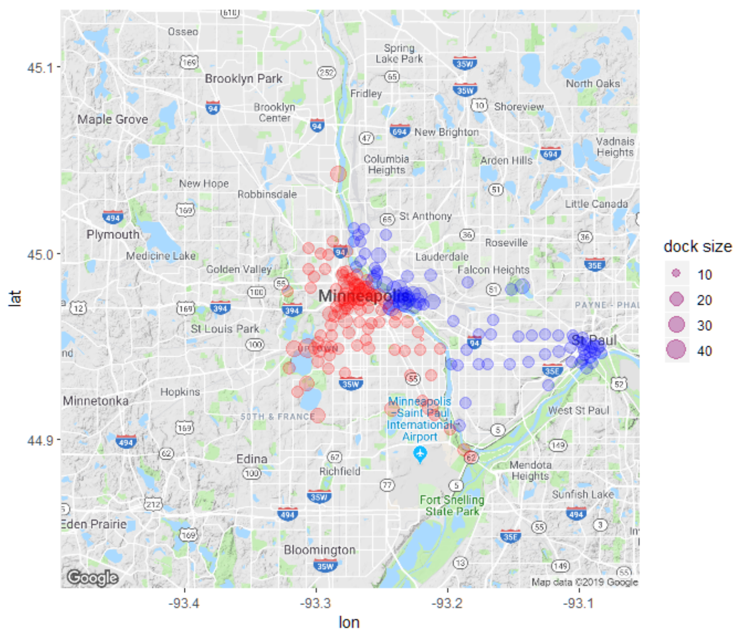


Figure 1: Nice Ride's station distribution & dock size

4 Case Study

We conduct numerical experiments under the real-life setting to examine the performance of the proposed two-threshold policy. In this section, we will explore the Nice Ride's data and the approaches to realize the upcoming demand pattern, and implement sensitivity analysis on several critical parameters.

4.1 Data Visualization

To complement the simulation study, we consider the real-world application of rental bike in twin cities. As one of the largest bike rental companies in twin cities, Nice Ride already builds up a network with more than 100 stations and above 1700 bikes among St.Paul and Minneapolis. Figure 1 shows how stations are distributed among the cities.

As you can see, most stations' capacities are around 20. Also, the areas with the highest bike station density are Minneapolis downtown and UMN's campus. Therefore, although treating the current bike system as a two-region system is too general, the proposed policy can still offer some contribution since most bike activities happen in those two areas. Hence, this project divides and regroups those bike stations by their zip codes into two broad groups -east (labeled with blue) and west (labeled with red). For the chosen Nice Ride rental record, it contained 95,147 trips over 45 weekdays and 20 weekends in 2018. The demand for bikes and docks in different time slot can be visualized in Figure 2.

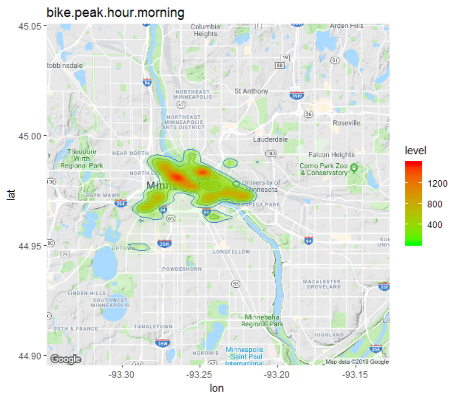
In Figure 2, time is divided into three slots -morning peak hour, evening peak hour and off-peak hour. The left and right three pictures are heat maps for the count of rental and return bikes happened in chosen days. The redder the location is, the more rental/return activities happen. From those graphs; we can observe precise time and directional features for the bike/dock demand: First, most of the bikes move from a spread residential area with lower bike station density to the concentrated workplace with higher bike station density in the morning (Figure 2-(a,b)). However, in the evening (Figure 2-(c,d)), the direction is opposite: more people rent bikes from a concentrated area and return them to spread residential areas. For the off-peak hour (Figure 2-(e,f)), there does not show a clear trend.

4.2 Simulation & Analysis

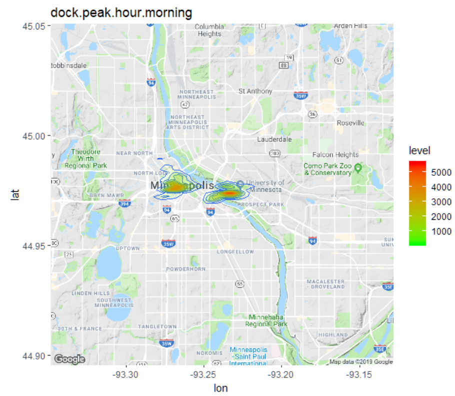
4.2.1 Simulation Settings

In the simulation, first we want to see how the change of dock/bike ratio impacts the two thresholds in the optimal policy. The parameters: return probability α, lost sales cost p, and the demand mean μ are estimated from the Nice Ride's data, while holding cost h, the repositioning cost s, and the capacity C are set manually by ourselves. The sensitivity analysis of the holding cost and repositioning cost are also provided in the following subsections. In this subsection we summarizes the settings for the simulation.

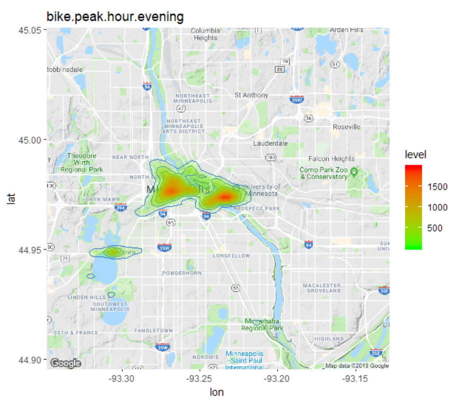
- Total period $T = 3$.
- Total number of bikes $N = 178$.
- The capacity ratio: $\frac{\text{total docks}}{\text{total bikes}} = 0.2, 0.4, \ldots, 2$. Balanced capacity for two locations is assumed.
- Lost sales cost
$$p = \begin{pmatrix} 0.8176 & 0.9412 \\ 0.9956 & 0.53 \end{pmatrix}$$
- Return probability:
$$\alpha_1 = \begin{pmatrix} 0.82 & 0.18 \\ 0.22 & 0.78 \end{pmatrix}, \alpha_2 = \begin{pmatrix} 0.85 & 0.15 \\ 0.19 & 0.81 \end{pmatrix}, \alpha_3 = \begin{pmatrix} 0.84 & 0.16 \\ 0.23 & 0.77 \end{pmatrix}$$
- Repositioning cost $s = 1.5$.
- Holding cost $h = 1$.
- Demand mean :
$$\mu = \begin{pmatrix} 12.02 & 31.25 & 22.94 \\ 7.81 & 29.61 & 16.54 \end{pmatrix}$$



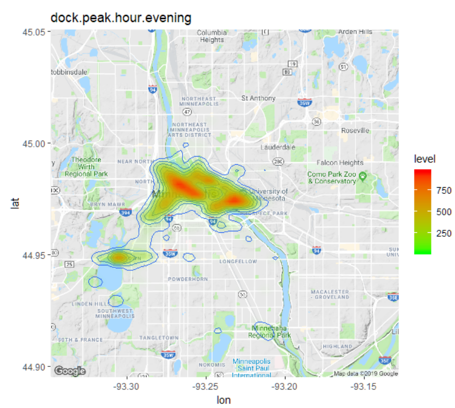
(a) Bike demand at morning peak hour



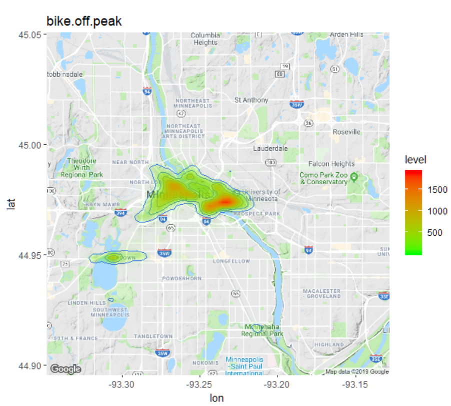
(b) Dock demand at morning peak hour



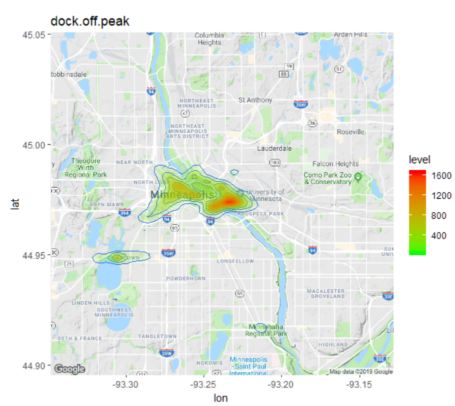
(c) Bike demand at evening peak hour



(d) Dock demand at evening peak hour



(e) Bike demand at off peak hour



(f) Dock demand at off peak hour

Figure 2: Heat maps for bike and dock demand at different hours

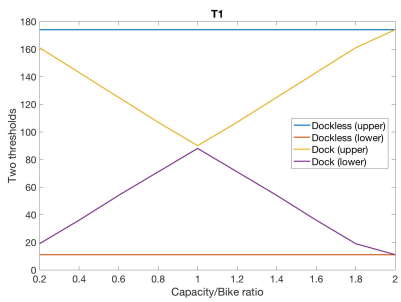
The total number of bikes N and the demand mean mu are scales down by a factor of 10 from the original estimated parameters from the data to reduce the simulation run time. The demand distribution is assumed to be uniformed in the interval $[0.5^*\mu, 1.5^*\mu]$.

4.2.2 Analysis: different dock/bike ratio

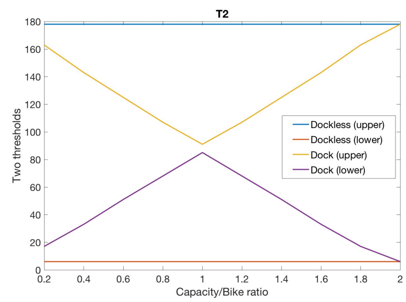
Figure 3 shows the results of optimal two thresholds and the optimal cost under different ratios of dock/bike. In Figure 3-(a) to 3-(c), first note that the two horizontal lines represent the two thresholds of the dockless system, and the region between the two lines is the no-repositioning region. The results show that there is almost no need for repositioning in the dockless system, which could result from the structure of the return probability α. In section 4.2.1 we see that α suggests that the most portion of rental will return to the same location, meaning that the flow of bikes between the two locations is few, resulting in a large no-repositioning region. In particular, the last period (Figure 3-(c)) shows that it's optimal not to reposition at all, which is due to the fact that it's the last period and there's no need to account for future cost.

From the results of dock-based system, we can see that when the dock number is either sufficiently low or high enough, the optimal two thresholds are the same as in the dockless system. In the former case, the event of not finding a dock to return (which we also refer to "out-of-capacity" event) is inevitable, thus the optimal thresholds won't be affected by the capacity constraint while it only increases the optimal cost. In the latter case the system is close to the dockless system (with ratio$\geq$ 2 corresponds to the dockless system), so again the optimal policy won't change. On the other hand, when the total number of docks is approximately equal to the total number of bikes (ratio=1), there are most opportunities to avoid out-of-capacity event by repositioning, therefore the corresponding no-repositioning region is the smallest. We can see that the similar results in the last period showing that there's less need of repositioning.

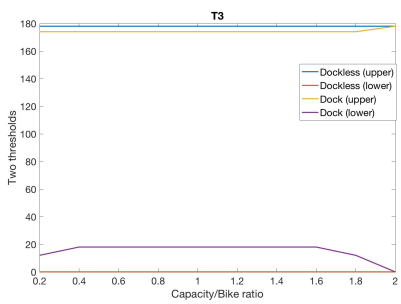
Since we model the effect of capacity constraint to be additional cost in the objective function without adding the cost of setting up the dock, it is natural to have the result in 3-(d), showing that the fewer docks in the system, the more holding cost is incurred and the higher the total optimal cost. While if we have enough docks, then the system resembles the dockless one and have the same optimal cost. However, if we add the cost for the docks in the objective function, for example, as a linear funciton of the dock number, then we can see that there's an optimal dock/bike ration to achieve the lowest cost. Although the dock-based system has a higher optimal cost by adding the holding cost, it's noteworthy that there might be certain disadvantages for the dockless system, for example it could be harder for the customer to locate a bike in such system, that we didn't take into account in our model. To compare the performance of these two systems on a fairer basis, one needs to incorporate these disadvantages for the dockless system.



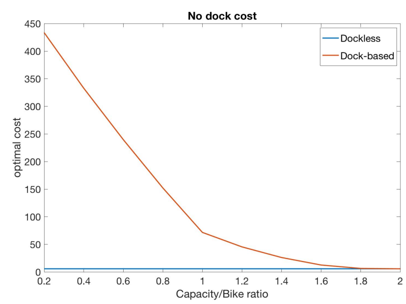
(a) Two thresholds: First period



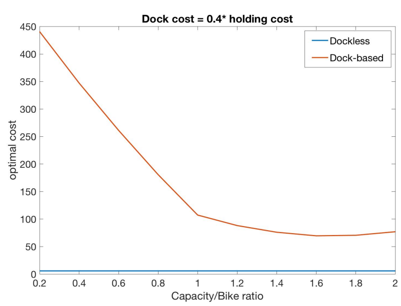
(b) Two thresholds: Second period



(c) Two thresholds: last period



(d) Optimal cost: no dock cost



(e) Optimal cost: with dock cost

Figure 3: Results for different dock/bike ratio

4.2.3 Sensitivity Analysis: holding cost

In the sensitivity analysis for holding cost, the capacity ratio (total docks/total bikes) is fixed at 1.4. The holding cost ratio varies across the range $0.5, 0.6, \ldots, 2$, corresponding to the holding cost $0.5 * 1, 0.6 * 1, \ldots, 2 * 1$ respectively. The lost sales cost is also increased by two times comparing to that in section 4.2.1, to ensure the assumption (lost sales cost outweighs the sum of average holding cost and repositioning cost) holds.

From the results for two thresholds in Figure 4, it is clear that higher holding cost results in smaller no-repositioning region in the dock-based system, since the system is more devoted to avoid out-of-capacity events. Moreover, the thresholds in the earlier period is more sensitive to the increase of holding cost, while in the last period the change of thresholds isn't significant until the ratio gets to 1.5, again demonstrate the tendency of not to reposition in the later periods.

Figure 4-(d) shows that the increase in optimal cost due to the increase of holding cost is limited, this is perhaps because of the assumption that the lost sales cost needs to outweigh the average holding cost and repositioning cost. The assumption makes sure that the effect of increasing holding cost is bounded by the lost sales cost. To further increase the holding cost, one needs to also increase the lost sales cost to ensure the assumption holds, so that the effect of lost sales cost should be more significant.

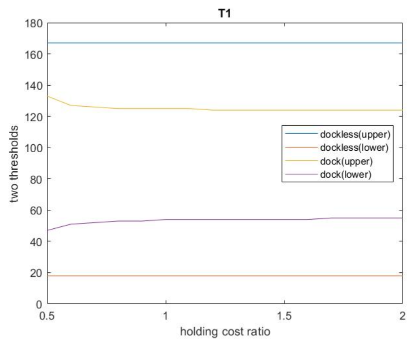
4.2.4 Sensitivity Analysis: repositioning cost

In the sensitivity analysis for different repositioning cost, the capacity ratio is again fixed at 1.4, while the ratio for repositioning cost varies from 0.5 to 2, which corresponds to the repositioning cost 0.5*1.5 to 2*1.5. The lost sales cost is again increased by factor 2 to ensure the assumption holds.

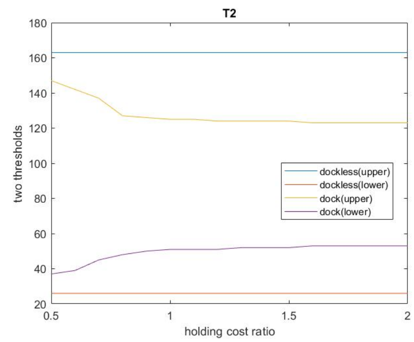
Figure 5 shows that the increase of repositioning cost results in larger no-repositioning region as expected. As opposed to the results in sensitivity analysis for holding cost, the earlier period is less sensitive to the increase in repositioning cost, since it needs to account more for the future cost regardless of the repositioning cost. On the other hand, the last period responds quickly to the increase in repositioning cost and tends to not reposition at all once the cost gets sufficiently high. In figure 5-(d) it shows that the optimal cost is more sensitive for the dock-based case, as we can expect from the results in section 4.2.2 of the analysis for the different capacity ratios. We know that the system with capacity constraint tends to reposition more than the dockless counterpart, thus increase in repositioning cost should have a more significant effect on such system.

4.2.5 Sensitivity Analysis: unbalanced capacity

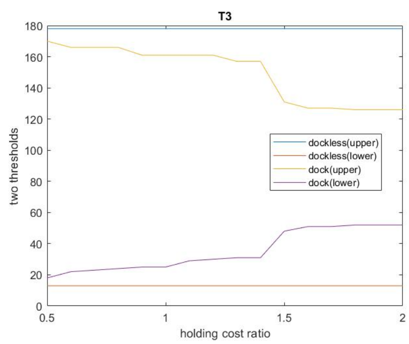
In previous simulation settings we assume balanced capacity between two locations (they have equal number of docks). In this section we aim to look into the



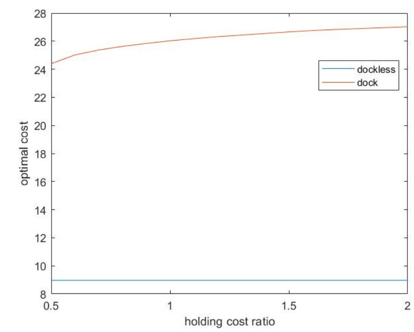
(a) Two thresholds: First period



(b) Two thresholds: Second period



(c) Two thresholds: last period



(d) Optimal cost: no dock cost

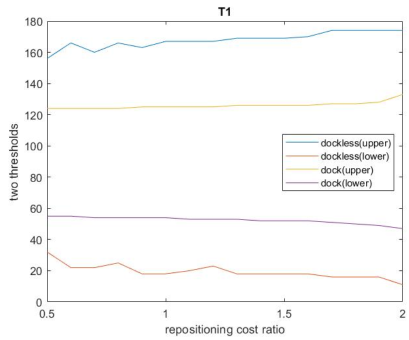
Figure 4: Sensitivity analysis for different holding cost

effect from unbalanced capacity. The capacity ratio is fixed at 1.4, and we let the ratio C_1/C_2 to varies from 0.5 to 2.

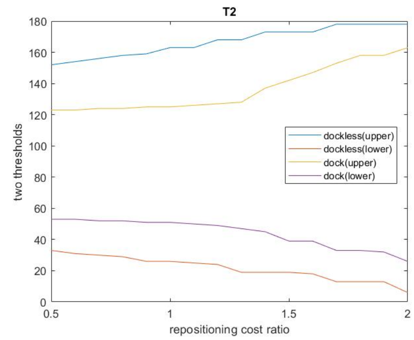
Figure 6-(a) to (c) shows that the size of no-repositioning region doesn't change with the unbalanced capacity ratio C_1/C_2. This is not surprising since the total capacity stays the same. However, as the ratio C_1/C_2 gets higher, meaning that there are more docks in location 1, the no-repositioning region shifts upward, indicating that it's optimal to have more inventory in location 1. This could prevent potential out-of-capacity events in location 2. Figure 6-(d) shows that the optimal cost doesn't change much due to unbalanced capacity, since the total capacity is fixed. However it does show that the balanced capacity results in lowest optimal cost.

5 Conclusion

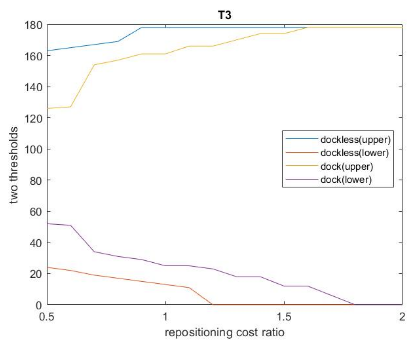
The project serves as an initial step to analyze the impact of capacity constraints on the repositioning policy in bike-rental system. The project focuses on the simplified problem setting where only two regions are considered, and



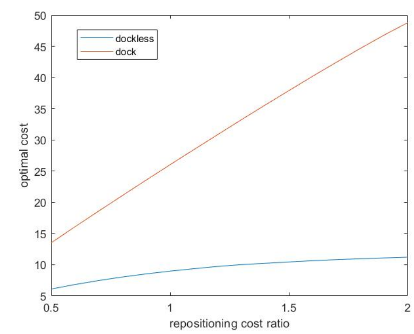
(a) Two thresholds: First period



(b) Two thresholds: Second period



(c) Two thresholds: last period

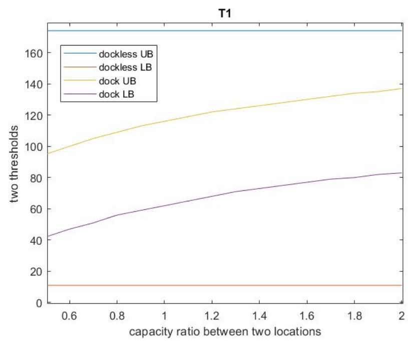


(d) Optimal cost: no dock cost

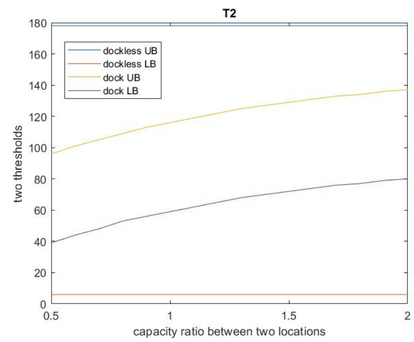
Figure 5: Sensitivity analysis for different repositioning cost

the penalty for not having an empty dock for return is modeled as a per unit holding cost. In this case, the optimal policy structure shows that it remains a two-threshold structure as in the dockless case. However, the threshold itself does change when the dock/bike ratio changes in the system. In short, the capacity constraint for the dock-based system creates more incentive to reposition the inventory properly. It is consistent with the intuition since the purpose of inventory-repositioning is to control the number of bikes at each station in a certain range (the no-repoisition region) where there's neither too many nor too few bikes in each station. The capacity (docks) can be viewed as the complement of the inventory (bikes), since too many bikes implies too few docks. This means that in the dock-based model the system in fact punishes unbalanced inventories twice, comparing to the dockless system. The result is reflected by shrinkage of the no-repositioning region as seen in the simulation results.

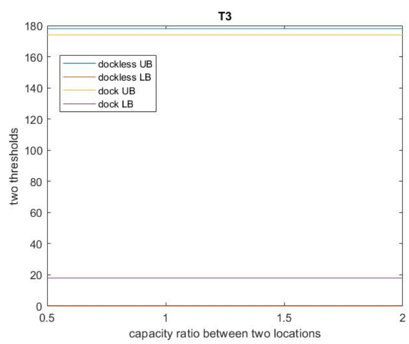
Due to the large state/action space when modeling such product-rental system with dynamic programming, the implementation of the derived policy is limited to small problem size, as in both Benjaafar et al. (2018) and He et al. (2018). However, in real world scenarios the problem often consists of much



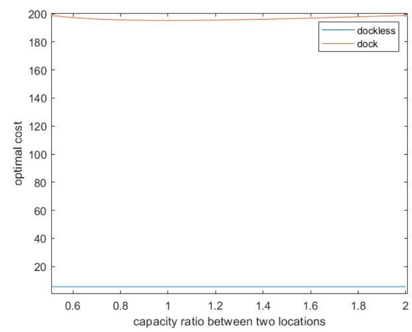
(a) Two thresholds: First period



(b) Two thresholds: Second period



(c) Two thresholds: last period



(d) Optimal cost: no dock cost

Figure 6: Sensitivity analysis for unbalanced capacity between two locations

more locations in a region than the algorithm can handle. Two possible solutions to this issue are proposed in Benjaafar et al. (2018): service region segmentation and hierarchical aggregation. These methods allow the algorithm to be implemented on a smaller scale of the regions. Nonetheless, certain issues could arise in the aggregation. For example, the lost sales and out of capacity happen in a station-wise scale, while through aggregation such events could be cancelled out. Therefore, the design of good aggregation method could be a future work along this line.

Finally, nowadays such bike-rental system has gradually evolved into dockless system. Therefore, using dock-based bike-rental system as a motivating example for the capacity-constrained model may not be relevant. On the other hand, electrical vehicles have become more popular in these few years and it's possible that they replace traditional vehicles in the future. Electrical vehicles require charging stations, which is a limited resource. This could potentially serve as a better motivating example in such product-rental system with capacity constraint, with the capacity being the limited charging stations, and the out-of-capacity cost could be modeled as not getting charged. As another potential future work direction, the operator not only can design a repositioning policy but also design how to motivate the customers to move the vehicles for the system, in order to achieve the benefit of inventory repositioning.

References

[1] Benjaafar, Saif and Jiang, Daniel and Li, Xiang and Li, Xiaobo, *Inventory Repositioning in On-Demand Product Rental Networks* (December 18, 2018). Available at SSRN:
`https://ssrn.com/abstract=2942921 or http://dx.doi.org/10.2139/ssrn.2942921`

[2] He, Long and Hu, Zhenyu and Zhang, Meilin,*Robust Repositioning for Vehicle Sharing* (March 31, 2018). Forthcoming in Manufacturing Service Operations Management. Available at SSRN:
`https://ssrn.com/abstract=2973739 or http://dx.doi.org/10.2139/ssrn.2973739`

[3] Wang, S., He, T., Zhang, D., Shu, Y., Liu, Y., Gu, Y., Liu, C., Lee, H., Son, S.H. (2018). *BRAVO: Improving the Rebalancing Operation in Bike Sharing with Rebalancing Range Prediction.* IMWUT, 2, 44:1-44:22.

Appendices

A MATLAB Code for Dockless Model

```
function [upper,lower,opt_val] = two_region(alpha,p,s,T,
    mu,N)

mu1 = mu(1,:);
mu2 = mu(2,:);
d = cell(2,T);
for k=1:T
    d{1,k} = floor(0.5*mu1(k)):floor(1.5*mu1(k));
    d{2,k} = floor(0.5*mu2(k)):floor(1.5*mu2(k));
end

p_bar = zeros(2,T);
for k=1:T
    p_bar(:,k) = sum(alpha{1,k}.*p,2); % average lost
        sales cost
end


action = zeros(N+1,T); % level after repositioning (y)
```

```matlab
value = zeros(N+1,T+1);

for t=T:-1:1
    for x=0:N
        re=x-N:x; % possible repositioning decision
        re_cost = zeros(1,numel(re));
        re_cost(re<0) = -re(re<0)*s(2,1);
        re_cost(re>=0) = re(re>=0)*s(1,2); % repositioning cost

        y = x*ones(1,numel(re)) - re; % inventory level after repositioning
        [D1, D2] = meshgrid(d{1,t},d{2,t});
        D1 = D1(:);
        D2 = D2(:);
        level_after = repmat(y,numel(D1),1); % y
        D1 = repmat(D1,1,numel(re));
        D2 = repmat(D2,1,numel(re));
        w1 = level_after+min(0,D1-level_after);
        %w1(level_after>=D1) = D1(level_after>=D1);
        w2 = D2+min(0,N-level_after-D2);
        %w2(w2>=D2) = D2(w2>=D2);
        x_next = level_after - floor(alpha{1,t}(1,2)*w1) + floor(alpha{1,t}(2,1)*w2);
        E_J = sum(p_bar(1,t)*(D1-w1)+p_bar(2,t)*(D2-w2)+reshape(value(x_next+1,t+1),size(x_next)),1);

        [value(x+1,t),action_index] = min(re_cost+E_J*(1/numel(d{1,t}))*(1/numel(d{2,t})));
        action(x+1,t) = y(action_index);
    end
end
upper = action(end,:)';
lower = action(1,:)';
opt_val = value(:,1);
end
```

B MATLAB Code for Dock Based Model

```matlab
function [upper,lower,opt_val] = two_region_dock(alpha,p,s,hold,T,mu,N,C)

mu1 = mu(1,:);
mu2 = mu(2,:);
d = cell(2,T);
for k=1:T
```

```
    d{1,k} = floor(0.5*mu1(k)):floor(1.5*mu1(k));
    d{2,k} = floor(0.5*mu2(k)):floor(1.5*mu2(k));
end


p_bar = zeros(2,T);
for k=1:T
    p_bar(:,k) = sum(alpha{1,k}.*p,2); % average lost sales cost
end

action = zeros(N+1,T); % level after repositioning (y)
value = zeros(N+1,T+1);

for t=T:-1:1
    for x=0:N
        re=x-N:x; % possible repositioning decision
        re_cost = zeros(1,numel(re));
        re_cost(re<0) = -re(re<0)*s(2,1);
        re_cost(re>=0) = re(re>=0)*s(1,2); % repositioning cost

        y = x*ones(1,numel(re)) - re; % inventory level after repositioning
        [D1, D2] = meshgrid(d{1,t},d{2,t});
        D1 = D1(:);
        D2 = D2(:);
        level_after = repmat(y,numel(D1),1); % y
        D1 = repmat(D1,1,numel(re));
        D2 = repmat(D2,1,numel(re));
        w1 = level_after;
        w1(level_after>=D1) = D1(level_after>=D1);
        w2 = N-level_after;
        w2(w2>=D2) = D2(w2>=D2);
        x_next = level_after - floor(alpha{1,t}(1,2)*w1) + floor(alpha{1,t}(2,1)*w2);
        hold_cost = hold(1)*max(0,x_next-C) + hold(2)*max(0,N-x_next-C);
        E_J = sum(p_bar(1,t)*(D1-w1)+p_bar(2,t)*(D2-w2)+hold_cost+reshape(value(x_next+1,t+1),size(x_next)),1);

        [value(x+1,t),action_index] = min(re_cost+E_J*(1/numel(d{1,t}))*(1/numel(d{2,t})));
        action(x+1,t) = y(action_index);

```

```
        end
end
upper = action(end,:)';
lower = action(1,:)';
opt_val = value(:,1);
end
```