공학박사 학위논문

A Dynamic Rebalancing Strategy in Public Bicycle Sharing Systems Based on Real-Time Dynamic Programming and Reinforcement Learning

실시간 동적 계획법 및 강화학습 기반의

공공자전거 시스템의 동적 재배치 전략

2020년 8월

서울대학교 대학원

공과대학 건설환경공학부

서 영 현

A Dynamic Rebalancing Strategy in Public Bicycle Sharing Systems Based on Real-Time Dynamic Programming and Reinforcement Learning

지도교수 고 승 영

이 논문을 공학박사 학위논문으로 제출함
2020년 5월

서울대학교 대학원
공과대학 건설환경공학부
서 영 현

서영현의 공학박사 학위논문을 인준함
2020년 6월

위 원 장 이 창 원 (인)
부위원장 고 승 영 (인)
위 원 김 동 규 (인)
위 원 추 상 호 (인)
위 원 강 승 모 (인)

Abstract

The public bicycle sharing system is one of the modes of transportation that can help to relieve several urban problems, such as traffic congestion and air pollution. Because users can pick up and return bicycles anytime and anywhere a station is located, pickup or return failure can occur due to the spatiotemporal imbalances in demand. To prevent system failures, the operator should establish an appropriate repositioning strategy. As the operator makes a decision based on the predicted demand information, the accuracy of forecasting demand is an essential factor. Due to the stochastic nature of demand, however, the occurrence of prediction errors is inevitable.

This study develops a stochastic dynamic model that minimizes unmet demand for rebalancing public bicycle sharing systems, taking into account the stochastic demand and the dynamic characteristics of the system. Since the repositioning mechanism corresponds to the sequential decision-making problem, this study applies the Markov decision process to the problem. To solve the Markov decision process, a dynamic programming method, which decomposes complex problems into simple subproblems to derive an exact solution. However, as a set of states and actions of the Markov decision process become more extensive, the computational complexity increases and it is intractable to derive solutions. An approximate dynamic programming method is introduced to derive an approximate solution. Further, a reinforcement learning model is applied to obtain a feasible solution in a large-scale public bicycle network.

It is assumed that the predicted demand is derived from the random forest, which is a kind of machine learning technique, and that the observed demand occurred along the Poisson distribution whose mean is the predicted demand to simulate the uncertainty of the future demand. Total unmet demand is used as a key performance indicator in this study.

In this study, a repositioning strategy that quickly responds to the prediction error, which means the difference between the observed demand and the predicted demand, is developed and the effectiveness is assessed. Strategies developed in previous studies or applied in the field are also modeled and compared with the results to verify the effectiveness of the strategy. Besides, the effects of various safety

buffers and safety stock are examined and appropriate strategies are suggested for each situation.

As a result of the analysis, the repositioning effect by the developed strategy was improved compared to the benchmark strategies. In particular, the effect of a strategy focusing on stations with high prediction errors is similar to the effect of a strategy considering all stations, but the computation time can be further reduced. Through this study, the utilization and reliability of the public bicycle system can be improved through the efficient operation without expanding the infrastructure.

Keywords: Markov Decision Process, Public bicycle sharing system, Real-time dynamic programming, Reinforcement learning, Repositioning

Student Number: 2014-21505

Contents

Chapter 1. Introduction 1

1.1 Research Background and Purposes 1

1.2 Research Scope and Procedure 7

Chapter 2. Literature Review 10

2.1 Vehicle Routing Problems 10

2.2 Bicycle Repositioning Problem 12

2.3 Markov Decision Processes 23

2.4 Implications and Contributions 26

Chapter 3. Model Formulation 28

3.1 Problem Definition 28

3.2 Markov Decision Processes 34

3.3 Demand Forecasting 40

3.4 Key Performance Indicator (KPI) 45

Chapter 4. Solution Algorithms 47

4.1 Exact Solution Algorithm 47

4.2 Approximate Dynamic Programming 50

4.3 Reinforcement Learning Method 52

Chapter 5. Numerical Example 55

5.1 Data Overview 55

5.2 Experimental Design 61
5.3 Algorithm Performance 66
5.4 Sensitivity Analysis 74
5.5 Large-scale Cases 76

Chapter 6. Conclusions 82

6.1 Conclusions 82
6.2 Future Research 83

References 86

초 록 92

List of Tables

Table 2.1 Summary of the static bicycle repositioning problem in the literature 16

Table 2.2 Summary of the dynamic bicycle repositioning problem in the literature 17

Table 2.3 Summary of the demand forecasting for the PBS system in the literature 21

Table 3.1 Chi-square test results for stations in Yeouido 33

Table 3.2 Relationship between desired service level and Z-score 38

Table 3.3 Descriptions of the variables for demand forecasting 43

Table 3.4 Mean decrease in accuracy for each variable of ST-73 45

Table 4.1 Algorithm of value iteration 49

Table 4.2 Algorithm of real-time dynamic programming 51

Table 4.3 Algorithm of actor-critic policy gradient 54

Table 5.1 Station-to-station travel time deployed in Yeouido 64

Table 5.2 Performance comparison between dynamic programming and RTDP 70

Table 5.3 Key performance indicators by benchmark strategies 72

Table 5.4 Key performance indicators by strategies 74

Table 5.5 Sensitivity analysis with varying Z-score and safety buffer 75

List of Figures

Figure 1.1 Number of full/empty instances by month of Capital Bikeshare system in 2014 3

Figure 1.2 Inventory variation of ST-9 in Seoul Bicycle Sharing System 3

Figure 1.3 Observed and forecasted pickup frequency (20 Sep 2017)..... 5

Figure 1.4 Location of stations and depot in Yeouido, Seoul..... 8

Figure 1.5 Research procedure..... 9

Figure 2.1 The repositioning period and forecasting period for SBRP 12

Figure 3.1 Prediction horizon in this study 30

Figure 3.2 Examples of observed return frequency and expected Poisson distribution: good-fit (upper) and bad-fit (lower) 34

Figure 3.3 The agent-environment interaction in a Markov decision process 35

Figure 3.5 Algorithm for the demand prediction..... 44

Figure 4.1 Relationship between number of stations and the number of states .. 49

Figure 4.2 Illustration of real-time dynamic programming..... 50

Figure 5.1 (a) The total number of pickups of bicycles, (b) daily pickup frequency heat map by day of the week and time of day 58

Figure 5.2 Relationship between meteorological factors and pickup frequency. 59

Figure 5.3 Daily pickup frequency by month (upper) and time of day (lower) of the PBS system in Seoul..... 60

Figure 5.4 Demand patterns for analysis period..... 63

Figure 5.5 Computation time needed for 10 iterations..... 71

Figure 5.6 Comparison with benchmark policies..... 73

Figure 5.7 Sensitivity analysis with varying Z-score and safety buffer 75

Figure 5.8 Performance analysis in deterministic demand context 77

Figure 5.9 Repositioning result of Strategy 1 78

Figure 5.10 Repositioning result of Strategy 2 79

Figure 5.11 Repositioning result by Strategy 3 79

Figure 5.12 Performance analysis in stochastic demand context 80

Chapter 1. Introduction

1.1 Research Background and Purposes

1.1.1 Public bicycle sharing system

As the development of ICT (Information Communication Technology) and the spread of smartphones enable real-time transmission and reception of data, the importance of a shared economy has been growing. A shared economy is based on collaborative consumption, in which produced products are shared by multiple people (Lessig, 2008). In terms of the efficient use of idle assets, the paradigm shifts from an era of individual ownership to an era of sharing goods or service, such as Uber, Airbnb and WeWork. The importance of a shared economy is expected to grow more and more due to the advantages of cost saving and convenient use, and services utilizing the concept are appearing continuously in every industrial sector. Typical examples of shared economy in the transportation sector are the car sharing system and the public bicycle sharing (PBS) system.

The PBS system, which contributes to alleviating urban problems such as traffic congestion and air pollution is a sustainable transportation mode that can meet last-mile traffic demands. After the introduction of the first generation of the PBS system in Amsterdam, Netherlands in 1968, many cities around the world have introduced the system (Shaheen et al., 2010). The number of cities operating a bicycle sharing system has increased from 13 in 2004 to 855 in 2014 (Fishman, 2016) and 1,608 systems were in operation and 391 prepared to be introduced as of June 2018 (Meddin, 2018).

In South Korea, the PBS system was first introduced in Changwon in 2008 with 20 stations and 430 bicycles (Shaheen et al., 2010) and since October 2015, a public

bicycle project (Ttareungyi) has been operated in Seoul, the capital city of South Korea. In the early days of the project, the number of users was low due to the concentration of the stations in a few areas, but the number increased sharply as the network expanded throughout Seoul. The system is so popular that Seoul citizens picked the Seoul Public Bicycle as the first place among the Seoul 2017 top-10 news (See http://english.seoul.go.kr/top-10-news-picks-2017-seoul-citizens). Despite the mountainous terrain in Seoul, users picked up an average of 4,400 bicycles per day in 2016 and the number increased to 27,000 bicycles per day in 2018. Because of the high utilization in Seoul, the city has expanded more stations and bicycles throughout the city.

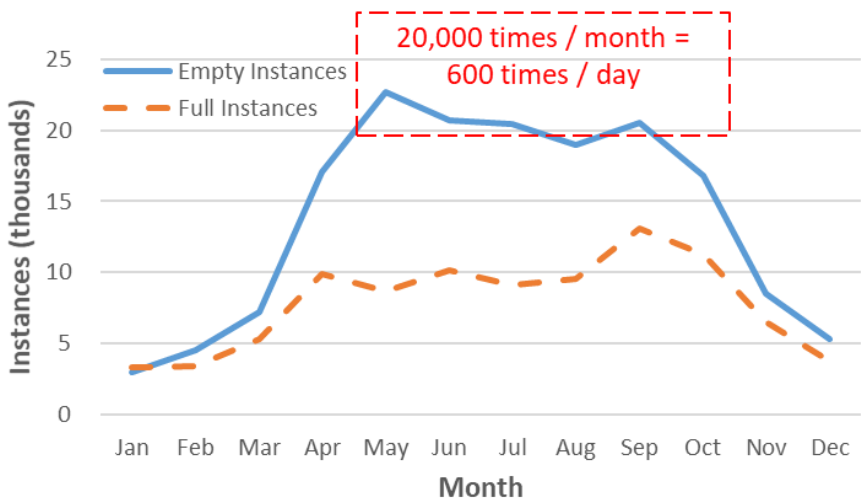
The service is popular because the PBS system has the advantage of allowing users to pick up and return bicycles wherever stations are located. It has been observed that public bicycles were used for commute trips to and from subway stations as well as for leisure trips in parks, indicating use for various trip purposes.

Accordingly, academic interest in public bicycle sharing systems is increasing in terms of planning and operating strategies (Nath and Rambha, 2019). Strategic planning includes areas such as a network design or the number of stations, station location or capacity determination. A typical example of operational planning is relocating bicycles.

1.1.2 Reposition of the public bicycle

Due to the random arrivals of PBS system users, a rapid change in the number of bicycles may result in an imbalance in station inventory. Figure 1.1 shows the number of empty and full instances of the PBS system in New York. It showed that about 20,000 times a month of empty state or more than 600 times a day on average. If pickup or return failure occurs repeatedly, the reliability of the system decreases. Figure 1.2 shows the inventory fluctuation of a station in the PBS system in Seoul

on 22 August 2017. If system failure repeats, users are unlikely to use public bicycles and they change their transportation mode. Therefore, to prevent the system failure, operators should establish a repositioning strategy. Most operators have employees deliver additional bicycles with trucks from stations where bicycles are plentiful to stations where more bicycles are needed.



Source: Capital Bikeshare (http://cabidashboard.ddot.dc.gov/CaBiDashboard/)

Figure 1.1 Number of full/empty instances by month of Capital Bikeshare system in 2014

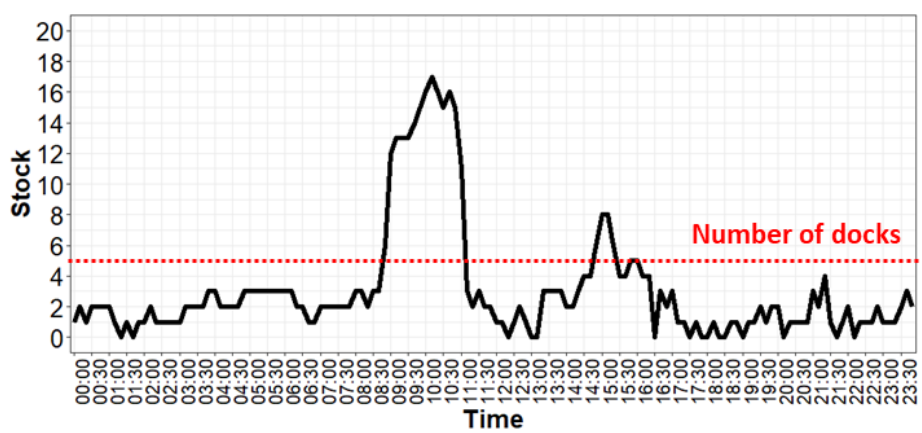


Figure 1.2 Inventory variation of ST-9 in Seoul Bicycle Sharing System

However, repositioning using trucks is limited by resources such as staff and

vehicles because the staffs should deal with the problems of bicycles that are broken or returned incompletely. Staff experience shows a tendency to move only to the shortest path or relocate bicycles from the current location to the nearest station. In the system in Paris, average bicycle usage was 110,000 bicycles per day, but only 3,000 bicycles were repositioned (Legros, 2019).

As the system size increases and a city becomes congested, the cost of repositioning public bicycles increases dramatically (Shin et al., 2012). As PBS systems are increasingly expanding, it is time to establish strategies to minimize repositioning costs. In addition, because the PBS systems in Korea were established by the public government, it is necessary to optimize the repositioning route in order to reduce costs and improve system efficiency.

A repositioning strategy aims to find an optimized route for the vehicle and to determine the optimal number of bicycles to load or unload for each station (Hagen and Gleditsch, 2018). To find an optimum number for bicycle distribution it is necessary to have accurate demand forecasting. Forecasting demand has been suggested as one of the challenges that a fourth-generation system must deal with (Shaheen et al., 2010). If the accuracy of the prediction is low, the safety stock needs to be increased to prevent the system failure. For example, Brinkmann et al. (2019) found that a repositioning strategy considering future demand was superior to the current strategy which focuses on deploying bicycles around nearby stations that have a shortage. This is because the bicycles are repositioned in advance to meet peak hour demand, reducing the unmet demands.

Due to external conditions or limitations of forecasting techniques, however, incorrect prediction inevitably occurs. Figure 1.3 shows the observed and the forecasted pickup demands of two stations, with one station not having a significant difference between the two values and the other having a potential error in distributing fewer bicycles due to the underestimated demand. Therefore, it is

necessary to respond to any errors that may occur in forecasting demand.

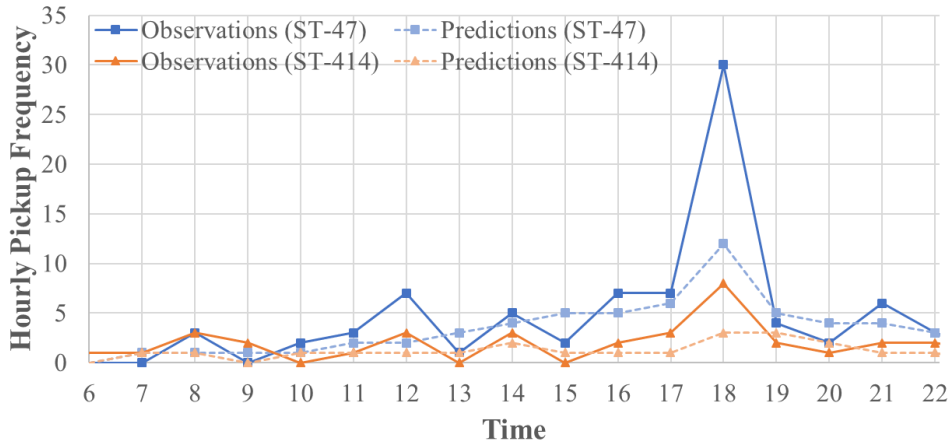


Figure 1.3 Observed and forecasted pickup frequency (20 Sep 2017)

1.1.3 Bicycle rebalancing problems as a sequential decision problem

The bicycle rebalancing problem can be represented as a sequential decision-making problem. After making a decision and observing the information, an agent makes more decisions and obtains more information. In other words, if the agent decides how many bikes to deliver or withdraw and the next station to move to, the system changes depending on the amount of the relocation and user demand. In the changed system, the agent makes a decision again, and the process of changing the system accordingly is repeated.

The sequential decision-making problem is classically formulated using a Markov Decision Process (MDP). MDP is a powerful analytical tool used for sequential decision making under uncertainty (Alagoz et al., 2010). It can lead to exact optimal policies in the long-run in a stochastic context (Legros, 2019). The stochastic nature of the system mandates that the rebalancing operation reacts to changing conditions in a timely manner (Kang et al., 2008). Solution methods of the

MDP include dynamic programming, evolutionary algorithm, or reinforcement learning.

Dynamic programming refers to a methodology to solve the problem by decomposing simple subproblems. It is well developed mathematically, but require a complete and accurate model of the environment (Sutton and Barto, 2018). When the size of state space and action space of the MDP increases, it is impossible to calculate the expected cost for all states and actions (curse of dimensionality). Therefore, dynamic programming has limitations in solving the problem and an approximate method should be considered for this system.

However, the complexity of the problem leads to a long time to solve the problem. The bicycle rebalancing problem has more things to consider than general VRPs. For example, the agent should identify the customers' inventory and the number of items to be loaded or unloaded from the vehicle. Therefore, it is necessary to have an algorithm that can solve the problem in a short time, and accordingly, most studies on rebalancing public bicycles have avoided use of the MDP methodology. Previous studies based on MDP simplified the problem, such as delivering bikes at a safety buffer margin or target fill levels or by visiting the nearest unbalanced station (Brinkmann et al., 2015). Stations are located throughout a city, but repositions are operated by zone. Therefore, problem decomposition considering repositioning context is required.

1.1.4 Research purpose

The purpose of this study is to develop a rebalancing model with stochastic demands and dynamic characteristics of PBS systems considering a fixed planning horizon. Stochastic means that demand is not known in advance and follows a stochastic distribution and dynamic means that subsequent decisions are made over a planning horizon (Brinkmann et al., 2019). As the demand fluctuates stochastically, there

occurs an error due to the difference between the forecasted and the observed demand and this leads to the necessity of repositioning by operators. For this purpose, the stochastic distribution of user demand is applied using historical data and dynamic programming is used. The effects of each strategy are evaluated according to various changes in conditions, such as network density and demand pattern. The performance of this model is compared with the performance of the strategies in the literature and greedy heuristics. Ultimately, policy implications are presented by proposing appropriate repositioning strategies for various situations.

1.2 Research Scope and Procedure

1.2.1 Research scope

The spatial scope of this study is Yeouido, Seoul in which there are 31 stations (Figure 1.4). Yeouido is one of the areas where the PBS system was launched in Seoul in 2015. There are business areas and parks, so the demand for commuters and park users is higher than that of other areas. The depot is also included in this study though it is located outside Yeouido because departure and arrival of the truck are made in the depot. The temporal scope is from August 2016 to September 2017, when pickup and return data could be obtained.

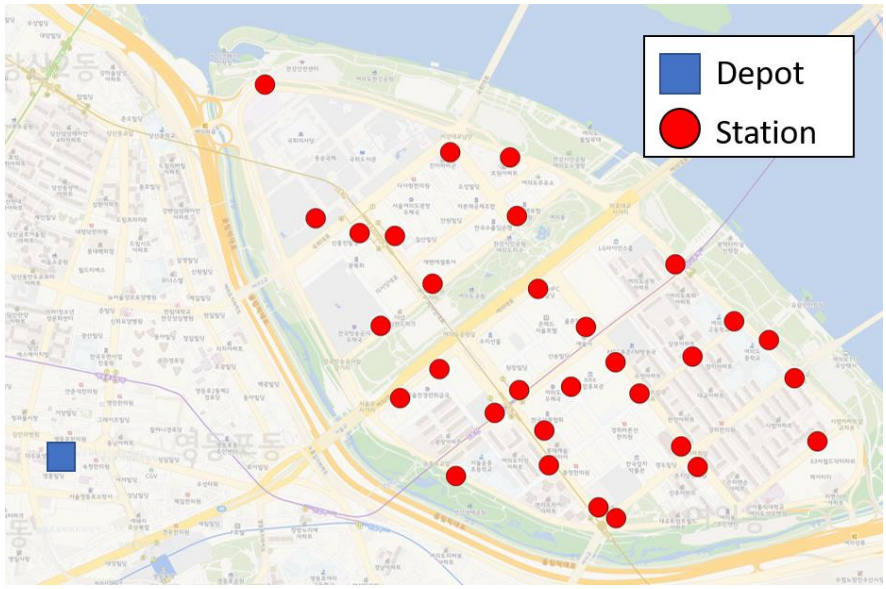


Figure 1.4 Location of stations and depot in Yeouido, Seoul

1.2.2 Research procedure

As shown in Figure 1.5, this study consists of literature review, model formulation, algorithm development, case study, discussions, and conclusions. Chapter 2 reviews the literature on PBS systems and especially repositioning issues. Chapter 3 describes the assumptions of this study, the model formulations, and MDP used in this study. Chapter 4 describes the algorithm used in this study such as real-time dynamic programming and reinforcement learning algorithm that were applied in this study. In Chapter 5, numerical examples are presented and the results are discussed. Data descriptions and descriptive statistics are also presented. Chapter 6 provides conclusions and ideas for future research.

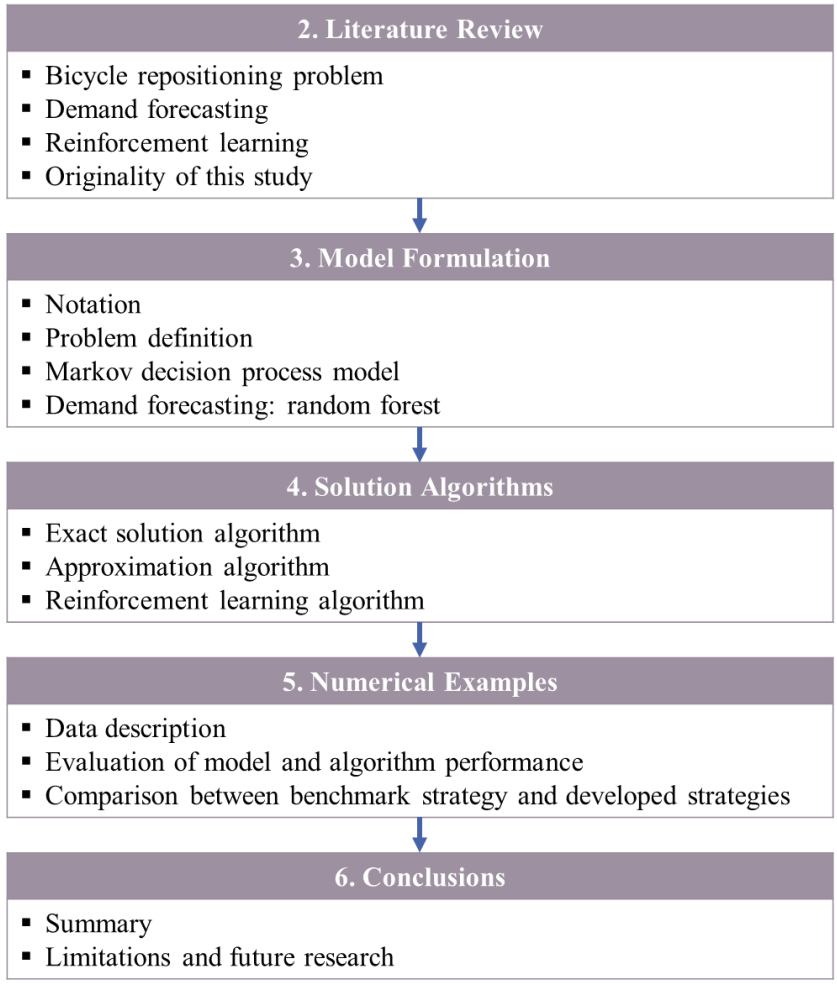


Figure 1.5 Research procedure

Chapter 2. Literature Review

2.1 Vehicle Routing Problems

A public bicycle repositioning problem is a vehicle routing problem (VRP) designing route of the repositioning vehicle, so the literature review begins with VRP. VRP was first introduced by Dantzig and Ramser (1959) as the Truck Dispatching Problem (Braekers et al., 2016). Subsequent subdivisions have been conducted, and more recently Mahmoudi and Zhou (2016) proposed the new time-discretized multi-commodity network flow model about vehicle routing problems with pickup and delivery with time windows (VRPPDTW); they allow the joint optimization of passenger-to-vehicle by incorporating the vehicle's status within the space-time transportation network.

VRP consists of various problems according to conditions and constraints. Rebalancing problem of PBS systems belongs to 1-PDTSP (one-commodity pickup and delivery traveling salesman problem), given that it is a problem of deriving the route to withdraw or distribute a single item, or a bicycle. Toth and Vigo (2014) described a bicycle repositioning problem as many-to-many problem since the public bicycle may have multiple origins and destinations and any station may be the origin or destination of the public bicycle.

2.1.1 Inventory routing problem

The inventory routing problem (IRP) deals with how suppliers deliver goods to customers within a given time. IRP integrates inventory management, vehicle routing, delivery-scheduling decisions (Coelho et al., 2014). Bell et al. (1983) first proposed the IRP to solve the cost minimization problem satisfying the customer inventory level under the stochastic demands.

A stochastic and dynamic inventory routing problem (SDIRP) is described in this section. Godfrey and Powell (2002) addressed a stochastic and dynamic resource allocation problem. A value function approximation (VFA) was used to anticipate potential future demand. A set of customers needs to be served over a set of days in Adelman (2004). For each day, a routing and inventory decision was determined. Bertazzi et al. (2013) applied a rollout algorithm (RA) to a SDIRP but RAs required a significant amount of runtime. Coelho et al. (2014) considered a problem that a route through a set of customers needs to be determined every day. This problem was deterministic based on average demand over a limited time horizon. Most SDIRP studies had limitations that continuously revealed demand has not been considered.

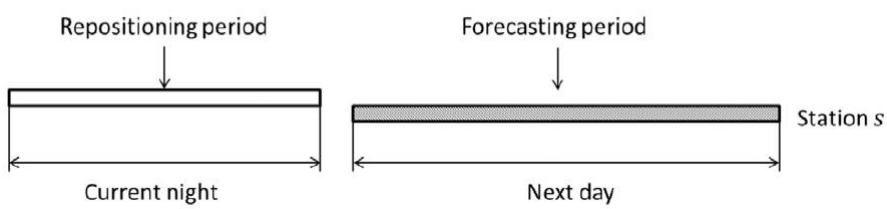
2.1.2 One commodity pickup-and-delivery TSP (1-PDTSP)

Mosheiov (1994) proposed a specified Travelling Salesman Problem (TSP), which exists pickup or delivery customers. Hernández-Pérez and Salazar-Gonzalez (2004) solved 1-PDTSP which minimizes traveling distance by applying the branch-and-cut algorithm, but there was a constraint that all nodes should be visited only once. Lei and Ouyang (2018) interpreted the repositioning issue of the PBS system as 1-PDTSP and used Continuous approximation (CA) approach. Hernández-Pérez et al. (2018) specified the repositioning problem of public bicycle as Split Delivery One Commodity Pickup-and-Delivery Travelling Salesman Problem (SD1PDTSP), and proposed a matheuristic algorithm that can solve the large-sized problem. SD1PDTSP is a problem that combines a capacitated vehicle routing problem (CVRP), a split demand vehicle routing problem (SDVRP), and 1-PDTSP. In their study, the maximum number of visits were limited to parameters and the station where there was no demand was also visited by trucks.

2.2 Bicycle Repositioning Problem

Research issues on PBS systems include the usage demand prediction, the repositioning strategies establishment including VRP, the incentive strategies to users, and the station location or capacity determination. Most studies have focused on the issues on the demand forecasting or the repositioning strategies of public bicycles.

Study on repositioning public bicycles is divided into two types, depending on the assumptions of when the operation is carried out. The first is the static bicycle repositioning problem (SBRP), which is assumed to ignore user activity as shown in Figure 2.1. This type can be regarded as an operation at night. As user demand increases during the daytime, however, there is a limit to just adjusting inventory after work hours (Zhang et al., 2017). In addition, demand often occurs randomly and user activity can change as a result of repositioning. In other words, although the system is actually a dynamic bicycle repositioning problem (DBRP) that changes over time, the complexity of the problem has led to the research on SBRP. Therefore, it is necessary to establish repositioning strategies that reflect forecasted demands to respond to changing inventories in real-time.



Source: Zhang et al. (2017)

Figure 2.1 The repositioning period and forecasting period for SBRP

The repositioning problem can also be classified as offline and online methods. Offline methods assume perfect knowledge of input data and do not react to changing

system states. Online methods react to the current inventory level and potentially other external factors. Most literature proposed use of a rolling horizon or an MDP and reinforcement learning framework. In this study, the previous studies are divided into SBRP and DBRP.

2.2.1 Static bicycle repositioning problem (SBRP)

Most of the studies on the bicycle repositioning problem focused on the SBRP. User demand is not comprised or assumed to be known in advance in the problem (Nath and Rambha, 2019). Chemla et al. (2011) proposed an exact algorithm based on the column generation. Erdoğan et al. (2015) constructed 1-PDTSP and branch-an-cut algorithm and solved the problem up to 60 stations within two hours with exact algorithm.

Raviv et al. (2013) proposed a penalty function that represented the expected number of shortages and included loading and unloading times within a time-constrained setting; they minimized the total cost of the system using mixed integer linear program (MILP).

Schuijbroek et al. (2017) designed optimal vehicle routes in terms of two aspects, determining the service level requirements at each station and designing optimal vehicle routes to balance the inventory.

Lin and Yang (2011) studied strategic design of public bicycle sharing systems with service level constraints; they proposed a formulation in which the penalty cost incurred by the unmet demand was added to the terms for the objective function, considering the number and the location of stations in the PBS system and the bicycle road network. The study assumed that future demand was fixed or followed the previous demand patterns and had no consideration on rebalancing bicycles.

Lin et al. (2013) formulated an objective function providing penalty costs associated with uncovered demand when considering the number and locations of

stations in the system and the network structure of bicycle lanes between stations. Ho and Szeto (2016) minimized the total travel cost incurred from visiting the nodes using greedy randomized adaptive search procedure (GRASP). Szeto et al. (2016) used the chemical reaction optimization (CRO) algorithm to minimize the weighted sum of the total number of unsatisfied customers and the vehicle's total operational time.

Although SBRP regards minimizing travel time (or distance) to be an essential factor, the problem has a limitation that the ultimate goal of the PBS system cannot be achieved in terms of its inability to respond after a daytime failure occurs. SBRP cannot handle non-recurring forms of demand fluctuations such as those due to weather or special events (Nath and Rambha, 2019).

2.2.2 Dynamic bicycle repositioning problem (DBRP)

DBRP focuses on minimizing unmet demand (or user dissatisfaction) that occurs during the repositioning process rather than on minimizing travel cost. As illustrated, most DBRP studies assumed a deterministic demand. Contardo et al. (2012) improved the computation time to solve the problem using a Dantzig-Wolfe decomposition and Benders decomposition and upper and lower bounds of the unmet demand were obtained. The assumption of the not time-varying demand was a limitation of the research.

Wang (2014) proposed a new mixed integer programming (MIP) model considering the dynamic characteristics of demand and compared the performance of the two heuristic solutions with an exact solution. The rolling horizon approach and Benders decomposition were applied to the study. Shui and Szeto (2018) introduced the environmental aspect of the PBS systems; they minimized the weighted sum of total unmet demand and total fuel and CO_2 emission cost using the artificial bee colony algorithm.

The assumptions in previous studies do not accurately reflect reality, as the predetermined route may be wrong due to the uncertainty in demand. More accurate estimates of demand for each station could reduce the inefficient movement of trucks and bicycles. Therefore, demand forecasting is the most basic and fundamental step for establishing dynamic repositioning strategies.

Zhang et al. (2017) developed an integrated model for forecasting inventory level, forecasting demand, repositioning, and routing; and they allowed employees to visit a station up to one time within the time window. Hagen and Gleditsch (2018) simplified and approximated the problem into a deterministic subproblem that assumed the known demands and the column generation heuristics were applied to the problem.

Fernández et al. (2018) presented four dynamic strategies: keeping inventory high, keeping inventory rates high, considering travel distances with inventory or inventory rates, and taking inventory of neighborhood stations into account together. Chiariotti et al. (2018) presented a strategy that first modeled the station inventory rate and determined the repositioning time and then selected the vehicle route and stations.

In a dynamic system, consideration needs to be given to dynamic characteristics that can be changed through decision making. In the public bicycle system, the static characteristics are nearby stations, the number of docks, the average number of pickups, or demand variation. The dynamic characteristics are inventory, the number of loading or unloading bicycles, repositioning route, or prediction error.

Table 2.1 Summary of the static bicycle repositioning problem in the literature

Reference	**Objective**	**Algorithm**	**Stochasticity**	**Dynamism**	**Number of stations**
Ho and Szeto (2014)	Minimize the total penalty cost	Tabu search	X	X	400
Erdoğan et al. (2015)	Minimize the total travel cost	Combinatorial Benders' cut	X	X	59
Dell'Amico et al. (2016)	Minimize the travel cost	Destroy and repair, branch-and-cut	X	X	564
Ho and Szeto (2016)	Minimize the total travel cost incurred from visiting the nodes	Greedy randomized adaptive search procedure (GRASP)	X	X	454
Szeto et al. (2016)	Minimize the weighted sum of unmet demand and the vehicle's operational time	Chemical reaction optimization (CRO)	X	X	300
Ho and Szeto (2017)	Minimize the weighted sum of the penalty cost and total travel time	Hybrid large neighborhood search (H-LNS)	X	X	518
Tang et al. (2019)	Minimizes the total penalty cost (upper-level model); minimizes the travel cost (the lower-level model)	Iterated local search and tabu search	X	X	20~200

Table 2.2 Summary of the dynamic bicycle repositioning problem in the literature

Reference	**Objective**	**Algorithm**	**Stochasticity**	**Dynamism**	**Number of stations**
Contardo et al. (2012)	Minimize the unmet demand	Dantzig-Wolfe decomposition, Benders decomposition	X	X	100
Zhang et al. (2017)	Minimize the total vehicle travel costs and the expected user dissatisfaction in the system	Heuristic algorithm	O	O	200
Shui and Szeto (2018)	Minimize the weighted sum of total unmet demand and total fuel and CO_2 emission cost	Artificial bee colony	X	O	180
Chiariotti et al. (2018)	Minimize the probability of the chance that a user experiences a service failure	Heuristic algorithm	O	O	280
Hagen and Gleditsch (2018)	Minimize the total violations, the total deviation, and the reward given for initiating trips	Column generation	O	O	158
Brinkmann et al. (2019)	Minimize the expected amount of unmet demand	Dynamic lookahead policy	O	O	169
Legros (2019)	Minimize the long-run overall rate of arrival of unsatisfied users	Dynamic programming	O	O	30

2.2.3 Relocation problem in other sharing systems

The study on the rebalance of one-way carsharing systems is relatively older than the study on the rebalance of the PBS systems. In this study, the scope of the review of the study on the car-sharing system is limited to station-based and staff-based relocation.

Proactive methods prepare the system for the expected future demand (Barth and Todd, 1999; Repoux et al., 2019) For example, reservation information is used to estimate the expected demand losses due to vehicle and spot shortages. On the other hand, active methods mean the shortest time and inventory balancing (Kek et al., 2006; Kek et al., 2009). The shortest time means that staff moves vehicles to or from a neighboring station in the shortest possible time. Inventory balancing implies filling a station that has a shortage of cars with a vehicle from another station which has an oversupply of cars.

A dynamic model means that the operation is executed successively at every event (Nourinejad and Roorda, 2014). This model is similar to the dynamic case in the PBS system, which is to find the optimal relocation and to find the corresponding relocation times (i.e., when to relocate a vehicle).

Unlike the PBS system, the one-way car sharing system has a characteristic to make reservations in advance, so this information can be used to rebalance systems. On the other hand, the PBS systems are not generally reserved, so demand forecasting is essential to reposition bicycles.

2.2.4 Demand Forecasting

Research on forecasting demand for shared public bicycles has been conducted for about ten years, and most of the studies have been published in the last five years. This section summarizes the contents of Seo et al. (2020).

In the past, traditional methods have been used to forecast demand. The conventional method such as multivariate linear regression was shown not to be proper for bicycle demand forecast (Feng and Wang, 2017). With the accumulation of abundant data and the development of machine-learning techniques, machine learning is currently being used to predict the demand for public bicycles. Many studies have used temporal factors such as hour, day, month, weekday, and holidays as well as meteorological factors such as temperature, precipitation, and wind speed to predict the demand for public bicycles.

Rudloff and Lackner (2014) proposed three ways to respond to a lack of demand: increasing the size of the system or the stations where there occurs regular full or lack events, repositioning with incentive, or repositioning using employees. The study developed demand models for pickups and returns for the Citybike Wien system in Vienna. They used count models, such as Poisson, negative binomial, and hurdle models. They considered meteorological factors as influences on demand and showed that the introduction of new stations was important in modeling the demand function.

Parikh and Ukkusuri (2015) suggested optimal inventory levels at the stations of a PBS system. Inventory levels were calculated for the stations that minimized the total penalty for the system after the penalty functions were estimated. Fournier et al. (2017) developed an estimation method of the monthly average daily bicycle counts and the average annual daily bicycle counts using a sinusoidal model to fit the typical pattern of seasonal bicycle demand. To develop the models, they used data from bicycle sharing systems in four cities and 47 permanent bicycle counters in six cities. However, this study was not appropriate for predicting daily fluctuations in demand.

Singhvi et al. (2015) predicted the demand of the bike-sharing system in New York by focusing on the morning peak during weekdays, with the use of taxis,

weather, and spatial variables as covariates. The study showed that aggregating stations in neighborhoods could improve the accuracy of the predictions. Rixey (2013) studied the effect of demographic and built-environment characteristics on the bicycle sharing system in Washington, D.C., Minneapolis-Saint Paul, and Denver in the United States.

Yang et al. (2016) suggested a spatio-temporal mobility model of bicycles based on historical bicycle sharing data and devised a traffic prediction mechanism based on station and time. Based on more than 100 million pickup records, the mobility model showed high prediction accuracy. 30-minute weather data (temperature, dew point, pressure, humidity, visibility, wind direction, wind speed, and conditions) were combined. The results of the evaluation showed an 85th-percentile relative error of 0.6 for predicting both pickups and returns. Regue and Recker (2014) addressed the station's activity, which was the standard deviation of the number of bicycles at the station during the last six intervals.

Since pickup and drop-off properties are different for each station, it is necessary to set the demand forecasting frequency concerning these characteristics. Some stations require frequent prediction, while other stations are enough to apply a modest prediction cycle. Most previous studies also have predicted future demand by considering temporal and meteorological factors. Under the same conditions, however, different demand patterns can appear. The number of activities in the previous time periods is required to detect this trend earlier. Faghih-Imani and Eluru (2016) analyzed the effect of time lag variables (one hour, one day, and one week before) on the arrival and departure rates, but the computation complexity required to take advantage of 1-hour before information in real-time was not considered.

Table 2.3 Summary of the demand forecasting for the PBS system in the literature

Reference	**Research Site**	**Timespan of Data**	**Demand Level**	**Model**	**Variables**
Rudloff and Lackner (2014)	Vienna, Austria	3 years (2010-2012)	• Station-level	Poisson, Negative binomial, Hurdle	• Weather • Full or empty neighboring stations on demand
Parikh and Ukkusuri (2015)	Antwerp, Belgium	1 year (unknown)	• Station-level	Negative binomial	• Starting inventory level at the station
Regue and Recker (2014)	Boston, U.S.	3 months (2012)	• Station-level	Linear regression, Neural networks, Gradient boosting machines	• Weather • Time • Station activity
Rixey (2013)	Washington, D.C., Minneapolis-Saint Paul, and Denver, U.S.	6-8 months (2010-2011)	• Station-level	Linear regression	• Demographic factors • Built environment factors • Transportation network factors
Singhvi et al. (2015)	New York, U.S.	1 month (2014)	• Station-level • Neighborhood-level	Linear regression	• Taxi usage • Weather • Spatial factors
Fournier et al. (2017)	Boston, Washington, D.C., New York, and Saint Paul, U.S.	3-5 years (2010-2015)	• Station-level	Regression	• Time • Number of bicycles

Froehlich et al. (2009)	Barcelona, Spain	13 weeks (2008)	• Station-level	Bayesian network	• Time • Number of bicycles • Prediction window
Lin et al. (2018)	New York, U.S.	3 years (2013-2016)	• Station-level	Graph Convolutional Neural Network with Data-driven Graph Filter	• Spatial distance • Demand • Average trip duration • Demand correlation
Yang et al. (2016)	Hangzhou, China	1 year (2013)	• Station-level	Random forest	• Weather

Source: Seo et al. (2020)

2.3 Markov Decision Processes

2.3.1 Markov Decision Processes

As described in Section 1.1.3, the bicycle rebalancing problem can be represented as a sequential decision-making problem. A relocation staff person determines the number of bikes to load or unload according to the current system state and moves to the next station where repositioning is required. As a result, more users can pick up or return bicycles and the staff repeats the same process according to the transitioned environment.

An MDP model can represent this type of problem. The model is composed of five factors: state, action, transition probability, reward, and discount factor. In the model, the set of actions, the rewards, and the transition probabilities depend only on the current state and action and not on states occupied and actions chosen in the past (Puterman, 2014). MDPs can formalize dynamic programming and reinforcement learning problems.

Research on the relocation of bicycles using MDP is rare and has recently begun to be studied. Legros (2019) tried to minimize the long-run rate of unmet demand and analyzed the case of a single vehicle and a time horizon that was segmented into periods of equal length without considering a predefined route.

Brinkmann et al. (2019) developed a dynamic lookahead policy (DLA) heuristic and showed that the RA could not obtain competitive results within a reasonable calculation time. The inventory decision was made to minimize unsatisfied demand at the current station within the horizon, and the routing decision was made to select the station that could most prevent unsatisfied demand within the horizon. The horizons per hour were determined by non-parametric VFA. The study was limited by the lengthy window period (1 hour).

2.3.2 Dynamic programming

Dynamic programming, proposed by Richard Bellman, is a method of solving a complex problem by breaking it down into simpler sub-problems in a recursive manner. The method has been widely used in many real fields such as transportation, finance, resource allocation (Powell, 2011). The shortest path problem is a well-known example of dynamic programming in a transportation network.

A mathematical form that describes the decision problem at each stage is named Bellman equation (Hamilton-Jacobi equation). The Bellman equation is as follows:

$$V(x_t) = \max[F(x_t, x_{t+1}) + \beta V(x_{t+1})]$$

where, $V(x_t)$: value function at state x at stage t

$F(x_t, x_{t+1})$: cost from x_t to x_{t+1}

β: discount factor

Dynamic programming calculates value function for all states. When the size of a state space and an action space of a model increases, it is impossible to calculate the expected cost for all states and actions. The state elements and action elements defined in the relocation problem of PBS system are more numerous than the general VRPs. For example, Brinkmann et al. (2019) considered timestep, stations' fill levels, vehicle's current station, and vehicle load as elements of the state space. The action space included an inventory decision and a routing decision in the study. As network size increases, calculation time using dynamic programming increases exponentially. For this reason, dynamic programming has not been used much in public bicycle relocation problems.

2.3.3 Reinforcement learning

Reinforcement learning method is a kind of learning method which inspires actions in response to the environment to maximize the agent's cumulative rewards in their interactions with the environment (Sutton and Barto, 2018). The agent does not have information which actions to take but should discover which actions provide the most reward through trial and error. The method was not commonly used much in transportation engineering field. Traffic signal control is the only field in transportation engineering that reinforcement learning method has been applied.

In terms of the sharing system, it consists of two approaches, a vehicle-based approach and a user-based approach. Li et al. (2018) proposed a clustering algorithm and a spatio-temporal reinforcement learning method for a vehicle-based approach. The clustering algorithm grouped stations and multiple trikes to reduce the problem complexity. The reinforcement learning model learned an optimal repositioning policy for each cluster, minimizing total unmet demand on a long-term horizon.

In a user-based approach, Pan et al. (2019) decided how to pay different users at each time, to incentivize them to help rebalance the system using a hierarchical reinforcement pricing algorithm with an MDP model. An objective function of the study was to maximize the total number of satisfied requests, subject to the rebalancing budget. The study considered the fill levels for each region, budget, previous pickup and return demand, previous expense, and past un-service rate as the state factors.

An et al. (2019) set an MDP problem with the goal of minimizing the cost of the car-sharing system. The study introduced two rewarding mechanisms, the picking bonus and the parking bonus to encourage users to balance the car-sharing system. The study used Deep Deterministic Policy Gradient (DDPG) method (actor-critic method).

2.4 Implications and Contributions

2.4.1 Implications

Based on a review of related work, although in reality the system changes over time, SBRP is mainly studied academically due to the complexity of the problem. Most DBRP studies considered deterministic demand and focused on minimizing the unmet demand during the repositioning process using the time-space network or MDP. Research using a time-space network is difficult to implement decision-making behavior including future information.

It is necessary to develop a model that simulates stochastic demands and dynamic programming for public bicycles. As the demands are stochastic and the in reality the system states are dynamic, stochastic dynamic programming for repositioning PBS systems is required. Other stochastic dynamic studies considered short-term strategies, but have not considered future demand (Brinkmann et al., 2015; Chiariotti et al., 2018). This strategy is similar to a simple heuristic. Even if the future demand is considered, only the next station to be visited is considered (Legros, 2019) or several target inventory levels are established (Brinkmann et al., 2019).

Moreover, it is necessary to develop a repositioning strategy that can cope with the inevitable emergencies caused by these dynamic characteristics. The strategy should be able to proactively respond to inventory shortages or excess that may occur due to inaccuracies in demand forecasts or rapid fluctuations in demand. In previous studies, the time unit of analysis was a lengthy time-window, so there is a limit to the detailed response at a peak time. Also, detailed information on demand distribution is lacking.

2.4.2 Contributions

The contributions of this study are as follows. This study develops an MDP based dynamic programming method that simulates the repositioning of the PBS systems with stochastic demand. Previous studies determined the agent's action in each state through simulation, but this study determines optimal actions in a given state through the proposed algorithm. An approximate dynamic programming (ADP) is developed to overcome the limitation of dynamic programming calculation time due to large state space and action space.

Reinforcement learning is also developed to apply the proposed algorithm to the real network. Future demand is predicted using the Seoul Bicycle Sharing system dataset. Little effort has been made to address the issue of relocating public bicycles using vehicles with reinforcement learning. Through the application to the real network, the implications of the proposed strategy and the policy implications of public bicycle relocation are presented.

In DBRP, a methodology for dealing with stochastic demand is a critical issue for problem-solving. The long-term strategy in this study can consider future stochastic demand. The prediction accuracy is improved by including the lag information of the number of pickups or returns in the demand forecasting model. A statistical distribution of demand is assumed through a statistical test based on historical demand data to generate stochastic demand.

This study develops a policy that effectively reduces the agents' action candidates, and derives implications through performance comparison for each policy. As the network size increases, the number of action candidates also increases, so a policy to effectively reduce action space is required. The action space can be reduced while proactively responding to an unexpected fluctuation of demand.

Chapter 3. Model Formulation

In this chapter, the model formulations are established for the development of dynamic repositioning strategies for the PBS system. This chapter describes definitions of sets and variables, problem definition, assumptions, model formulations, and key performance indicators.

3.1 Problem Definition

3.1.1 Notation

A notation used in this study is described below. The notation is mainly referenced from Brinkmann et al. (2019).

Sets

$N = \{n_0, \dots, n_{\max}\}$	Set of stations (0: depot)
$T = \{t_0, \dots t_{\max}\}$	Set of timesteps
$S = \{s_0, \dots, s_{\max}\}$	Set of states
$A_s = \{a_0, \dots, a_{\max} \mid a = (\iota^a, n^a)\}, \quad \forall s \in S$	Set of feasible actions
$\Pi = \{\pi_0, \dots, \pi_{\max} \mid \pi: S \to A\}$	Set of policies

Indices

k	Action point
$t_k \in T$	Point in time in state s_k
$s_k^a = (s_k, a), \ \forall s \in S, a \in A_s$	Post-action states

Parameters

c_v	Vehicle capacity
$\tau(\cdot,\cdot)$	Travel time between two stations
τ_r	Service time for relocation per bike
c^n	Station capacity
β	Safety buffer
z	z-score for the safety stock
p_k	Station observed pickup demand in time t_k
d_k	Station observed return demand in time t_k
$\hat{p}_k$	Station predicted pickup demand in time t_k
$\hat{d}_k$	Station predicted return demand in time t_k

Variables

f_k^v	Vehicle load in time t_k
$n_k^v \in N$	Vehicle location in time t_k
y_k	The number of delivered bicycles from vehicle in time t_k
$f_k = \left(f_k^{n_0}, \dots, f_k^{n_{\max}}\right)$	Station fill levels in time t_k
$i_k = (i_k^{n_0}, \dots, i_k^{n_{\max}})$	Station fill rate index in time t_k
ι^a	Delivery decision
n^a	Next station decision

3.1.2 Problem definition

Each station has an initial inventory f_0^n, a capacity c^n, and predicted pickup and return demand $\widehat{p_k^n}, \widehat{d_k^n}$ in the time interval $[t_0, t_{k_{\max}}]$. The depot is assumed to have an enormous capacity and no demand. The observed pickup demand p_k and return demand d_k at timestep t_k are not known in advance. At timestep t_k, observed demands p_{k-1}^n and d_{k-1}^n are revealed respectively and inventory f_k^n is changed by the observed demands and delivery. An agent, a vehicle with loading capacity c_v, should start at depot n_0 and return to the depot at the end of the time horizon.

$$k = k_{\max} \iff A_{s_k} = \{(\iota^{s_k}, n_0)\}$$

The agent determines the number of bikes to be delivered or withdrawn at the current station and the next station to visit at every decision point. If pickup demand or return demand is not satisfied for each station due to lack of bicycles or docks, an unmet demand occurs.

The aim of the problem in this study is to find a vehicle route and the number of bikes to deliver at stations so that the sum of the weighted sum of the expected unmet demand and the travel time is minimized.

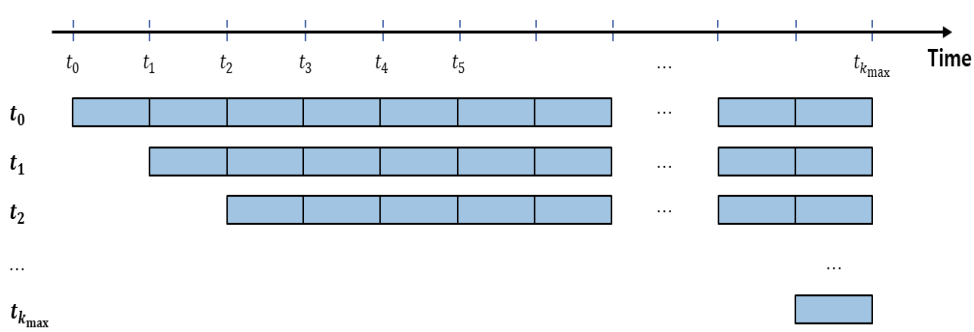


Figure 3.1 Prediction horizon in this study

3.1.3 Assumptions

To simplify the problem, several assumptions were made in this study. First, it is assumed that the trip of public bicycles has a spatiotemporal pattern, and that future demand follows the historical pattern. Based on this assumption, future demand can be predicted using historical demand. In addition, observed demands are assumed to follow a non-homogeneous Poisson distribution to present customers' random arrival processes. A detailed description of the Chi-square test to demonstrate this is provided in the next section.

A station is allowed to be visited at most once excluding a depot. This assumption is reasonable for two reasons. First, the agent distributes or withdraws bicycles to reach a number of safety stock at the station in this study. The safety stock means inventory that prevents future stockout, so a single visit can prevent out of stock within the horizon. The second reason is that the assumption can make the solution space smaller, making the development of an efficient algorithm to solve the problem much easier (Ho and Szeto, 2014; Raviv et al., 2013). A vehicle has a capacity of 15 bicycles and travels to stations by Euclidean distance at a speed of 20 km/h. Handling time is one minute per bicycle.

3.1.3.1. Chi-square test for demand distribution

A Chi-square goodness of fit test is performed on the return data to determine if the demand distribution for public bicycles follows a specific probability distribution function. The reason for using the return data is the characteristic of the PBS system in Seoul, which a bicycle can be returned unconditionally through connecting to another bicycle already returned. In other words, the return data is no censored data, so it is accurate to test the statistical distribution of the return demand. The time

period for the analysis was 10 minutes from 18:00 to 18:10, and the temporal range of this study is September, so the frequency of 10-minute return data from 17:50 to 18:20 on September weekdays after 2016 was analyzed.

The null hypothesis and the alternative hypothesis are as follows.

Null hypothesis (H_0): Return frequency follows the Poisson distribution.

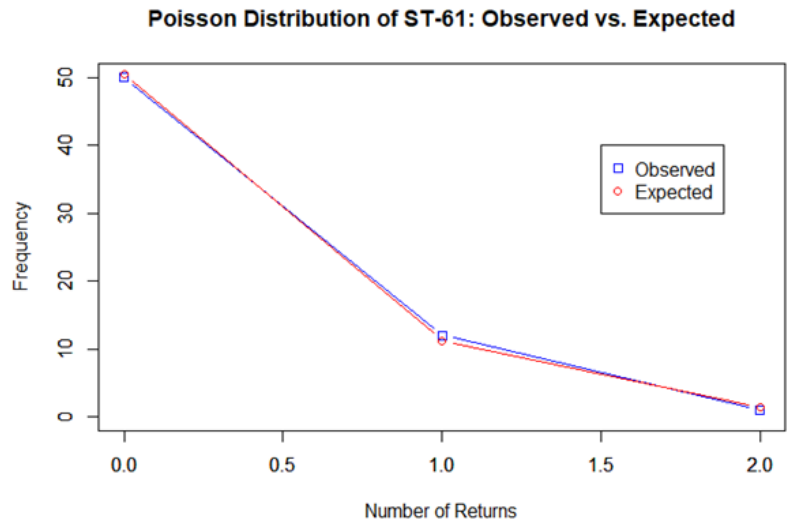
Alternative hypothesis (H_1): Not H_0

The formula for the test statistics is $\chi^2 = \sum_{i=1}^{k} \frac{(O_i - E_i)^2}{E_i}$ where k is the number of classes, O_i is an observed frequency, and E_i is an expected frequency.

As shown in Table 3.1, the p-values of only 7 stations were lower than 0.05 among 31 stations in Yeouido. In other words, it was found that return frequency at only 7 stations did not follow the Poisson distribution. The characteristic of these stations is that the return occurs frequently. Examples of stations with high p-value (ST-61) and low p-value (ST-73) are presented in Figure 3.2. A maximum of two returns have been recorded in 10 minutes at the ST-63, but ST-73 had a maximum of 14 returns. Based on the results of the Chi-square test, it is reasonable to assume that the demand in the model follows the Poisson distribution.

Table 3.1 Chi-square test results for stations in Yeouido

Station	**λ**	**χ^2**	**df**	**p-value**	**Station**	**λ**	**χ^2**	**df**	**p-value**
ST-45	0.143	0.053	1	0.818	ST-66	0.794	5.874	3	0.118
ST-46	0.381	2.314	3	0.510	ST-61	0.222	0.138	2	0.933
ST-47	0.762	21.401	5	0.001	ST-62	0.540	4.590	3	0.204
ST-51	0.365	0.188	2	0.910	ST-63	0.635	4.878	3	0.181
ST-50	0.429	2.009	2	0.366	ST-67	0.984	4.100	5	0.535
ST-52	0.397	1.157	3	0.763	ST-68	0.873	6.465	4	0.167
ST-53	0.476	7.184	3	0.066	ST-69	0.603	2.536	2	0.281
ST-73	4.492	108.784	12	0.000	ST-70	1.032	7.890	4	0.096
ST-55	0.444	4.535	2	0.104	ST-71	0.667	8.751	3	0.033
ST-56	0.762	2.806	4	0.591	ST-72	0.540	0.344	2	0.842
ST-57	2.270	18.682	7	0.009	ST-296	0.683	Inf	5	0.000
ST-58	1.063	11.826	5	0.037	ST-297	0.667	39.729	5	0.000
ST-59	0.381	0.282	2	0.868	ST-414	0.302	0.273	2	0.872
ST-60	0.302	3.829	3	0.281	ST-424	0.286	1.090	2	0.580
ST-64	0.381	16.911	3	0.001	ST-425	0.238	0.295	2	0.863
ST-65	0.937	8.438	5	0.134					



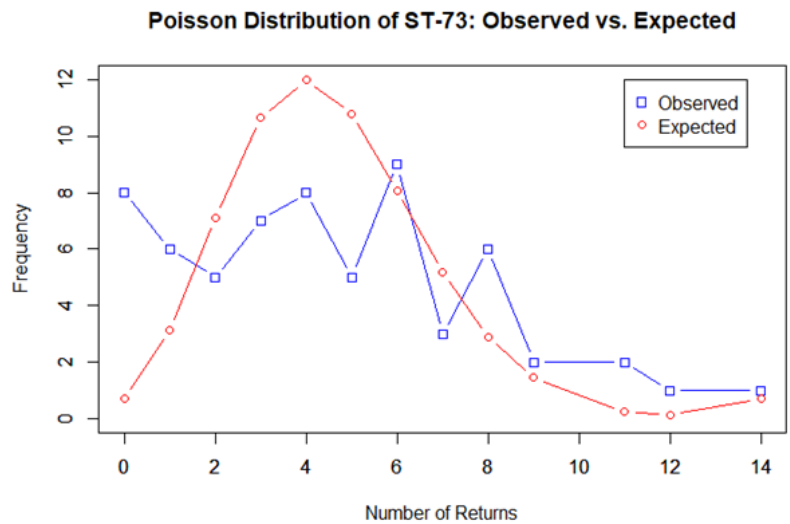


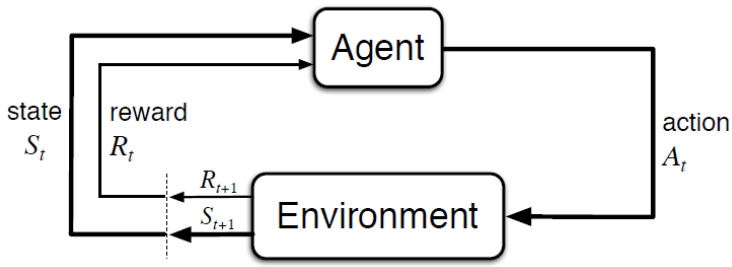
Figure 3.2 Examples of observed return frequency and expected Poisson distribution: good-fit (upper) and bad-fit (lower)

3.2 Markov Decision Processes

3.2.1 Concept

MDPs are based on the interaction of an agent and the environment (Figure 3.3). The agent makes a decision in a given state and the decision changes the environment.

The agent is given a reward and the next state information by the environment, which allows for the subsequent decision.



Source: Sutton and Barto (2018)

Figure 3.3 The agent-environment interaction in a Markov decision process

The MDP model can be represented by a five-tuple, (S, A, Pr, R, γ). State (S) represents the information of the entire environment at each moment. Action (A) is agent's action. Transition probability (Pr) is defined as the probability of transition from state s_t to state s_{t+1} when taking an action a_t. The solution of this problem is to find the optimal policy $\pi^* \in \Pi$ which describes the best action for each state in the MDP.

$$\pi^* = \underset{\pi \in \Pi}{\operatorname{argmin}} \mathbb{E}\left[\sum_{k=0}^{k_{max}} p\big(s_k, \pi(s_k)\big) | s_0\right]$$

A policy means a rule that determines a decision given the available information in state S_t (Powell, 2011). A policy is classified into deterministic policy and stochastic policy. A deterministic policy represents one action in a given state, and a stochastic policy represents the probability of each action. Each tuple is described in detail in the following sections.

The scheme of the PBS system can be represented by the MDP (Brinkmann et al., 2019; Puterman, 2014). In terms of rebalance to the PBS system, a fleet of trucks

serves as the agent and the system corresponds to the environment. The agent determines the number of bikes to deliver to the station from the vehicle or to withdraw from the station to the vehicle. A solution to the problem is the policy minimizing the expectation of the costs. Therefore, the objective is to find the optimal policy.

3.2.1.2. State

According to the study by Nath and Rambha (2019), states are typically comprised of inventory levels and locations of repositioning vehicles and their contents in the context of bike repositioning. The state space of this study is constructed with reference to Brinkmann et al. (2019). Three factors were included in the state space: (t_k, n_k^v, i_k). t_k is the timestep, n_k^v is the current station of the vehicle, and i_k are binary variables representing a station's fill rate index. i_k has a value of zero if the fill rate is between safety buffers, and one if otherwise. The safety buffer is defined as an interval of a certain percentage of capacity c^n, and the interval can be adjusted by β.

$$i_k = \begin{cases} 0, & \text{if } \beta c^n < f_k^n < (1-\beta)c^n \\ 1, & \text{otherwise} \end{cases}$$

The fill rate index is less accurate in indicating the station's information than the fill level, but the number of states can be significantly reduced by aggregating the fill level.

The number of the timesteps is $|T|$ and the vehicle can be located at any station. Each station has two values for the fill rate index, so the number of possible fill rate indices for all stations is $2^{|N|}$. Therefore, the dimension of the state space is $|S| \leq |T| \cdot |N| \cdot 2^{|N|}$.

3.2.1.3. Action

The agent's action at each decision point consists of two consecutive decisions, which is a delivery decision and a next station decision. First, the number of bikes to be loaded or unloaded at the current station is determined according to the target fill level of the station. Among the two actions, this study considers only the next station decision, while the delivery decision is automatically determined by external factors to reduce action space.

As the expected demand may fluctuate due to weather or incidents, safety stock is introduced. The safety stock is inventory that is carried to prevent stockouts (King, 2011). Stockouts stem from factors such as demand fluctuation or prediction inaccuracy. The safety stock equation is as follows. Under the assumption that demand follows a Poisson distribution, the standard deviation of demand can be replaced by the mean of demand. Also, total lead time (PC) and time increment used for calculating standard deviation of demand (T_1) are assumed to be the same.

$$(\text{Safety stock}) = z\sqrt{PC/T_1}\sigma_D$$

where, z: Z-score

PC: total lead time

T_1: time increment used for calculating standard deviation of demand

σ_D: standard deviation of demand

Table 3.2 shows the relationship between desired cycle service level and Z-score. The desired cycle service level means the percentage of preventing stockouts. Higher cycle service levels require disproportionately higher Z-scores.

Table 3.2 Relationship between desired service level and Z-score

Desired cycle service level (%)	**Z-score**
84	1
85	1.04
90	1.28
95	1.65
99	2.33
99.9	3.09

Source: King (2011)

First, the preliminary delivery decision δ_i is determined by the sum of future net demand and the safety stock taking into account the predicted demand. If the expected total pickup demand is higher than the total return demand, the current station should be in a condition where a bicycle with net expected pickups multiplied by the Z-score can be picked up. Conversely, if the return demand is expected to be higher than the pickup demand, the station should accept the bicycles with the net expected returns multiplied by z. z is a statistical figure known as a standard score.

$$\delta_i = \begin{cases} (1+z)\sum_{t_k}^{T}(\hat{p}_k - \hat{d}_k) - f_k^{n_k^v} & \text{, if } \sum_{t_k}^{T}(\hat{p}_k - \hat{d}_k) > 0 \\ (1+z)\sum_{t_k}^{T}\left(\hat{p}_k - \hat{d}_k\right) + \left(c^{n_k^v} - f_k^{n_k^v}\right), & \text{if } \sum_{t_k}^{T}(\hat{p}_k - \hat{d}_k) < 0 \end{cases}$$

If the sum of future net demand is zero, the target fill level is set to an amount by which the inventory becomes a safety buffer margin.

$$\delta_i = \begin{cases} \beta c^{n_k^v} - f_k^{n_k^v} & \text{, if } \quad f_k^{n_k^v} < \beta c^{n_k^v} \\ -\beta c^{n_k^v} + \left(c^{n_k^v} - f_k^{n_k^v}\right) & \text{, if } \quad f_k^{n_k^v} > (1-\beta) c^{n_k^v} \\ 0 & \text{, otherwise} \end{cases}$$

The actual delivery decision is affected by f_k^v and c_v. The number of bikes on the vehicle might be lower than the preliminary delivery decision, or the preliminary delivery decision might be higher than the number of vacancies of the vehicle. Therefore, the actual decision (e.g., the number of bikes to be delivered or withdrawn) is determined under the next constraints.

$$\iota_i = \begin{cases} \min(\delta_i,\ c^{n_k^v} - f_k^{n_k^v}, f_k^v)\,, & \text{if } \delta_i > 0 \\ \max(\delta_i, -f_k^{n_k^v}, f_k^v - c_v)\,, & \text{otherwise} \end{cases}$$

Agent's second action is the routing decision. All stations can be candidates as the next station to be visited at the next timestep. However, in general, other stations can be visited on the way to a station far away. In this study, the strategy of prioritizing stations to visit can reduce the size of the action space.

3.2.1.4. Reward

A reward is a value that the agent needs to determine the action. In this study, the reward is set as the weighted sum of total unmet demands from all stations given action a_k and realization of the transition $\omega: S \times A \to S$, and the total travel time of the vehicle. The reason for considering the travel time is to eliminate the contradiction in which the reward is the same if the failed demand is the same, even if different stations are selected as the next station. The agent moves according to the

policy that minimizes the reward.

3.2.1.5. Transition probability

A post-action state is changed by the agent's action and the users' pickup or return demand during the corresponding timestep. If the demand is deterministic and known in advance, then the post-action state is determined. The transition probability to the corresponding state is one, while the probabilities to the other states are zero. As a result, the calculation of the Bellman optimality equality becomes quite simple. In this study, however, the stochastic demand is considered, and the transition probability to the post-action state should be calculated.

The pickup demand and return demand are assumed to follow a time-dependent Poisson distribution. The Skellam distribution, which is a discrete probability distribution of the difference of two Poisson distributions with respective expected values, is applied for calculating the transition probability.

3.2.1.6. Discount factor

The discount factor means the reduction rate of reward over time. The closer it is to one, the more the value of the future reward will be treated equally to the present value. The reason the discount factor is important is that the current reward is usually more significant than the future reward.

3.3 Demand Forecasting

This section describes the demand forecasting method used in this study. The methodology was already used in our previous study (Seo et al., 2020) and the remaining of this section summaries the study.

3.3.1 Random forest technique

The random forest technique, an ensemble learning method used for classification and regression, was proposed by Breiman in 2001. The general idea of the method is to combine huge decision trees that are identically distributed and each decision tree is built individually built on a bootstrapped sample of data. The correlation between decision trees is reduced by generating identically distributed decision trees repeatedly, and this leads to the reduction of the dispersion of prediction errors. The predictions are performed by averaging the output values from each decision tree. This technique is a type of committee method and an improved technique of bagging, and it can obtain remarkable performance with little in terms of tuning. See Breiman (2001) and Hastie et al. (2009) for details of the technique.

Compared with other algorithms, the random forest model is more suitable for predicting public bicycle demand. First, it can deal with both categorical and numerical variables without normalization (Yang et al., 2016). This study regards the temporal factors as categorical values (year, season, month, day of the week, and hour) or binary variables (weekday and holiday), and the meteorological factors as continuous values (temperature, precipitation, and wind speed). Hence, the approach can be used without an additional quantification when coping with complex variables. Second, it provides the relative importance of the factors, which can give insights into the patterns of public bicycle use. For example, the hour factor generally has the most significant impact on the demand for bicycles in Seoul, which means that there are a lot of periodical trips such as commuting or going to school. Third, because the technique can deal with big data and execute computation faster, it is proper for modeling pickup or return behavior based on millions of trip data.

3.3.2 Model construction

Independent variables in demand forecasting were selected by reviewing previous studies and analyzing descriptive statistics. The descriptive statistics and the relationship between ridership and temporal and meteorological factors are discussed in detail in Section 5.1.3. In addition to these factors, station activity information was added to the variable set, which represents the number of pickups or returns at a station during the previous time on the day. The reason why the time lags on the day were considered is that the patterns are expected to change dynamically by the previous pickups or drop-offs on the day and that the usage patterns may vary as the meteorological factors are different on one day or one week ago. Table 3.3 provides the independent variables selected in this study.

3.3.3 Demand forecasting process

A demand forecasting process is illustrated in Figure 3.4. First, input data such as historical pickup data, weather data, and holiday data are built and preprocessed. The forecast unit is an hour and the number of pickups and returns are aggregated on an hourly basis. Demand prediction is conducted with the model constructed in the previous section. Hourly predicted demands are uniformly distributed in 10 minutes, which is the unit of the timestep in this study. For validation, observed and predicted demands are compared at each timestep and the prediction errors can be calculated.

Demand forecasting models were built from the historical data and the accuracy of the models should be evaluated. The historical data were divided into 70% training set and 30% test set. The data from the 1st to the 21st of each month were used as the training set, and the data from the 22nd through the last day of each month were set as the test set. The experiment was conducted with the ‘randomForest’ package of R.

Table 3.3 Descriptions of the variables for demand forecasting

Variable	**Description**	**Source**
Year	Year of the time	-
Season	Categorical variable representing the season (1-Mar to May, 2-Jun to Aug, 3-Sep to Nov, 4-Dec to Feb)	-
Month	Month of the time	-
Day of the Week	Categorical variable representing the day of week (1-Sun, 2-Mon, 3-Tue, 4-Wed, 5-Thu, 6-Fri, 7-Sat)	-
Hour	Hour of the time	-
Holiday	Dummy variable (0, 1) that indicates if a given day was an official holiday	Open Data Portal
Temperature	Average temperature in Celsius for the corresponding time	Korea Meteorological Administration
Precipitation	Hourly precipitation in millimeters	
Wind Speed	Average wind speed in meters per second	
Lag Information	Number of pickups or returns during the one hour ago, two hours ago, or three hours ago	Seoul Facilities Corporation

Source: Seo et al., (2020)

As mentioned by the features of the random forest model, the importance of input variables can be checked by using the varImpPlot function in randomForest (Breiman et al., 2018). For the station at which most pickups occurred (ST-73 at Exit 1 of Yeouinaru Station), the mean decrease in accuracy of each model was calculated and presented in Table 3.4. The mean decrease in accuracy is the value obtained by averaging the difference for the out-of-bag data of all decision trees between the prediction error after permuting each predictor and the prediction error before the permutation. The data after making a bootstrap sample of each decision tree is out-of-bag data. The higher values mean that the variable has greater importance. The 'hour' variable was analyzed as having the highest explanatory power, and the reason

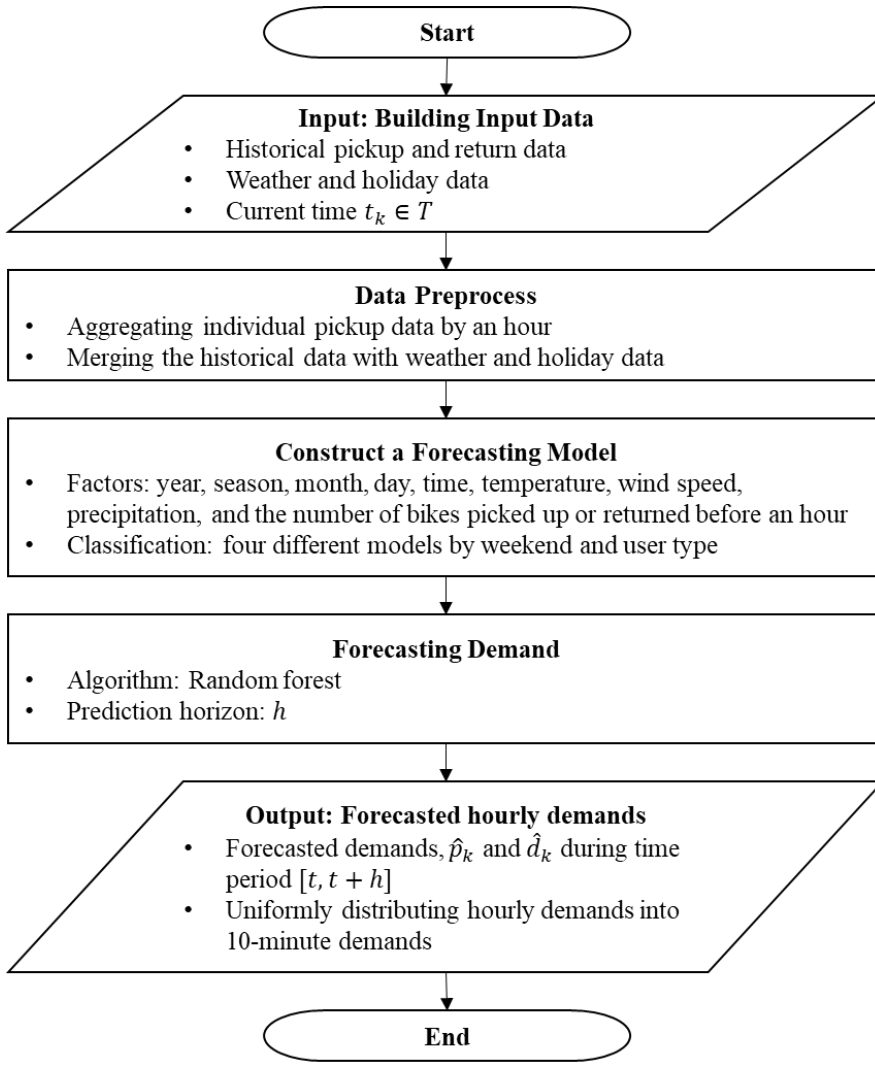


Figure 3.4 Algorithm for the demand prediction

for the high value of the 'year' variable was the continuous expansion of the PBS system in Seoul. Regarding lag information, the value when using the number of pickups one hour ago was the highest at 100, the number of pickups two hours ago was 66.6, and the number of pickups three hours ago was 54.0. In other words, the closer the lag information is to the present time, the better it explains future demand.

Table 3.4 Mean decrease in accuracy for each variable of ST-73

Variable	**With one hour ago information**	**With two hours ago information**	**With three hours ago information**	**Without lag information**
Year	109.9	119.1	126.0	108.3
Season	28.4	26.8	26.6	29.7
Month	42.1	44.0	43.5	37.7
Day of the Week	-1.7	-2.3	-3.2	-0.2
Hour	184.6	192.5	204.7	134.4
Temperature	41.0	40.2	41.7	43.9
Precipitation	13.3	12.0	16.0	19.9
Wind Speed	30.8	41.3	51.4	59.3
Lag Information	100.0	66.6	54.0	-

Source: Seo et al., (2020)

3.4 Key Performance Indicator (KPI)

In order to compare the effects of each strategy in this study, unmet demand is used as the key performance indicators (KPI).

3.4.1 Unmet demand

Pickup failure occurs when $p_k^n > f_k^n$ and return failure occurs when $d_k^n > (c^n - f_k^n)$. Unmet demand means the total number of failed pickup or return demands from all stations. The unmet demand is counted when user cannot pick up a bicycle due to the lack of bicycles or cannot return it due to the full of bicycles. Since the observed demand is not realized when determining the vehicle route, the route is derived using the forecasted demand. Thus, when observed demand is realized, there may be a higher observed pickup demand than expected, resulting in shortage of inventories.

There are alternative indices such as the number of repositioned bikes, satisfied

demand of the repositioned bikes, satisfied demands of the visited station, or unmet demands of the visited station. Travel time is not considered because it is already included as a trade-off between unmet demands and distance.

Chapter 4. Solution Algorithms

4.1 Exact Solution Algorithm

4.1.1 Dynamic programming

Dynamic programming is a method of solving a complex problem by breaking it down into simpler sub-problems in a recursive manner. This structure is based on the Principle of Optimality described by Bellman. In other words, an optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy about the state resulting from the first decision (Bellman, 1957).

The model-based method uses the Bellman equation. As shown in the below equation, every state has a value V. Dynamic programming finds an action that maximizes the sum of the rewards of the action and the expected value of possible next states and updates the value with the value of the corresponding state.

$$V_\pi(s) = \mathbb{E}_\pi[R_{t+1}(S_t, a_t) + \gamma V_\pi(S_{t+1})|S_t = s]$$

Dynamic programming can be used to compute optimal policies given a perfect model of the environment as an MDP (Sutton and Barto, 2018). The idea of dynamic programming is to construct the search to find the optimal policy using value functions. A value is defined as the expected long-term return of the current state under policy π.

4.1.1.1. Value iteration

Value iteration computes the Bellman optimality equation by dynamic programming.

Bellman optimality equation is as follows:

$$V_t(S_t) = \max_{a_t \in A}[R_t(S_t, a_t) + \gamma \mathbb{E}\{V_{t+1}(S_{t+1})|S_t, a_t\}]$$

where, $V_t(S_t)$: value function at state S_t,

$R_t(S_t, a_t)$: reward incurred by taking an action a_t at state S_t

γ: discount factor

The value iteration is virtually identical to backward dynamic programming for finite horizon problems (Powell, 2011). At each iteration, the estimate of the value function determines which actions we will make and as a result defines a policy. The estimate of the value function is updated for every state at each iteration.

The value iteration algorithm is represented in

Table 4.1 (Powell, 2011). The value function of all states is initialized and the value function of the terminal state is set to 0. In each state, the Bellman equation is calculated and the largest value is selected as the new value. The iteration stops at convergence, whenever Δ is smaller than a predetermined tolerance θ for all states.

The state space of dynamic programming is too large to enumerate all the space. The dimension of the state space is $|S| \leq |T| \cdot |N| \cdot 2^{|N|}$. Figure 4.1 shows the graph of the number of states according to the number of stations. As the number of stations increases, the number of states grows exponentially. For example, there are 10 stations (10 docks per each station) for repositioning 2 hours. The number of states is higher than 1.2×10^5, and the calculation is intractable. This result suggests that it is impossible to derive a solution with dynamic programming, which calculates the value function of all states. Therefore, an approximate algorithm is required for the repositioning problem of the PBS systems.

Table 4.1 Algorithm of value iteration

Algorithm: Value Iteration

Step 0. Initialization:

Initialize $V(s)$ for all states $s \in S$ arbitrarily

except that $V(terminal) = 0$

Set $\Delta = 0$

Step 1. Calculation:

for each state **do**

$v \leftarrow V(s)$

Calculate: $V(s) \leftarrow \max_{a \in A}\left(C(s,a) + \gamma \sum_{s' \in S} \mathbb{P}(s'|s,a)V(s')\right)$

Compute: $\Delta \leftarrow \max(\Delta, |v - V(s)|)$

Step 2. If $\Delta > \theta$, return to Step 1. Else stop.

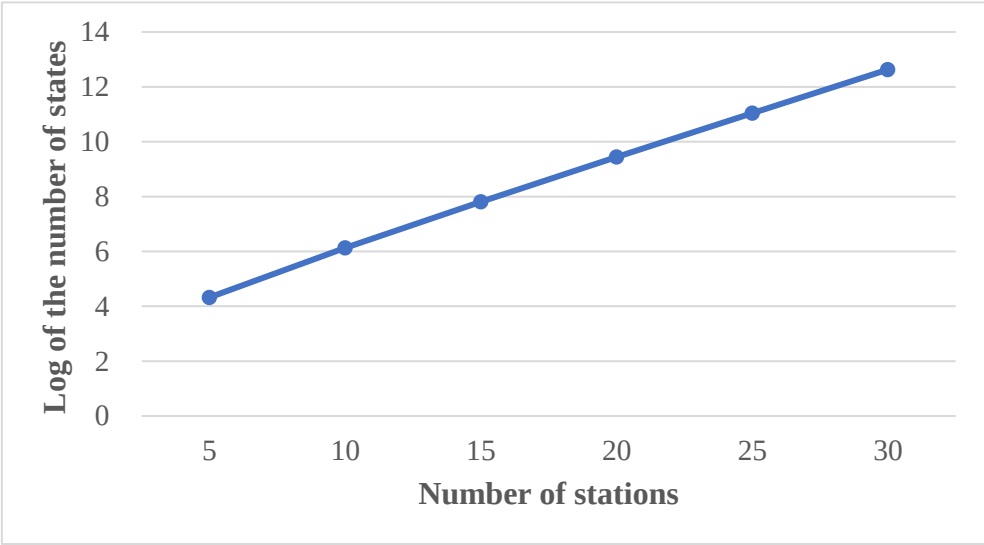


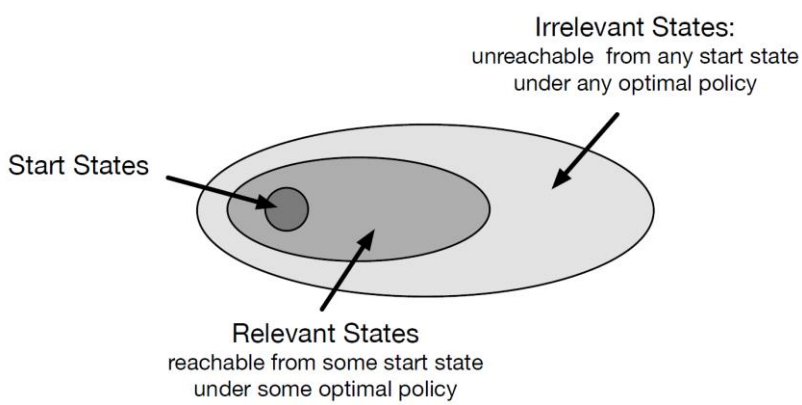
Figure 4.1 Relationship between number of stations and the number of states

4.2 Approximate Dynamic Programming

In this study, an asynchronous technique, Real-time dynamic programming (RTDP), is developed to derive an approximate solution. Asynchronous means not updating all states the same number of times, but updating some states once and some states multiple times.

4.2.1 Real-time dynamic programming

Real-time dynamic programming (RTDP) is proposed by Barto et al. (1995). The idea of RTDP is that an agent only visits states that are relevant to the agent (Figure 4.2). RTDP is an on-policy trajectory-sampling version of the value-iteration algorithm of dynamic programming (Sutton and Barto, 2018). RTDP updates the value of states visited in actual or simulated trajectories by means of expected tabular value-iteration updates. For certain types of problems satisfying reasonable conditions, RTDP is guaranteed to find a policy that is optimal on the relevant states without visiting every state infinitely often (Sutton and Barto, 2018).



Source: Sutton and Barto (2018)

Figure 4.2 Illustration of real-time dynamic programming

Required conditions for convergence of RTDP are as follows, according to Sutton and Barto (2018): 1) the initial value of goal state is zero, 2) there exists at least one policy that guarantees that a goal state is reached with probability one from the start state, 3) all rewards for transitions from non-ending states are strictly negative, and 4) the initial values of all states are equal to zero.

After selecting a sample of random demands, the Bellman optimal equation is solved. The corresponding state is only updated, and the rest are not updated. The agent moves to the next state s' according to the action and the sample demand and repeats the same process.

Table 4.2 Algorithm of real-time dynamic programming

Algorithm: Real-time Dynamic Programming

Step 0. Initialization:

Initialize $\bar{V}^0(s)$ for all states s.

Choose an initial state S^1

Set $n = 1$

Step 1. Choose a sample path ω^n.

Step 2a. Solve:

$$\hat{v}^n = \max_{a \in A^n} \left(C(s_k, a) + \gamma \sum_{s' \in S} \mathbb{P}(s'|s_k, a) \bar{V}^{n-1}(s') \right)$$

and let a^n be the value of a that solves the maximization problem.

Step 2b. Update $\bar{V}^{n-1}(S^n)$ using

$$\bar{V}^n(S) = \begin{cases} \hat{v}^n, & S = S^n \\ \bar{V}^{n-1}(S), & \text{otherwise.} \end{cases}$$

Step 2c. Compute $S^n = S^M(S^n, a^n, W(\omega^n))$.

Step 3. Let $n = n + 1$. If $n < N$, go to Step 1.

Even RTDP is a tabular method, so the computation is intractable when the state

space is enormous. Therefore, to obtain feasible solutions in a large-scale network, a way of approximating the value function is needed.

4.2.2 Manipulating algorithm

Due to the nature of the Bellman optimal equation, the possible next states and all actions should be considered. Two manipulations are possible to reduce computational effort. At first, the action space can be reduced. In terms of the routing decision, the agent considers only the stations that meet certain conditions as the next station to visit.

- Strategy 1: Consider all stations
- Strategy 2: Consider stations close to the current station
- Strategy 3: Consider stations with large forecasting errors

Second, the next state may vary due to the stochastic demand, but considering all states is inefficient. For example, if most stations have low predicted demand, the probability that $i_k = [0, \dots, 0]$ is transitioned to $i_{k+1} = [1, \dots, 1]$ is quite low. The possible fill levels of the following state are assumed to be within a standard deviation of the Skellam distribution, $\sqrt{\hat{p}_k + \hat{d}_k}$.

$$f_k + y_k - \sqrt{\hat{p}_k + \hat{d}_k} \leq f_{k+1} \leq f_k + y_k + \sqrt{\hat{p}_k + \hat{d}_k}$$

4.3 Reinforcement Learning Method

A feasible solution should be obtained in real-time even for the large-scale bike-sharing system. It is impossible to update and save a value function in a table form for all state-action pairs. Reinforcement learning is a method that enables agents to learn without prior knowledge about the environment and the model. Given different

rewards depending on the action, the agent tries to make a high reward action. Unlike DP and RTDP, the reinforcement learning approximates a value function using an artificial neural network (ANN) without storing it in a table.

4.3.1 Actor-critic method

Actor-critic technique is a combination of policy-based learning and value-based learning, and has two neural networks. Each of the two models respectively calculates an action based on the state (actor) and calculates the Q-value of the action (critic). Q-value is similar to value, except that it takes an extra parameter, the current action a. $Q_\pi(s,a)$ refers to the long-term return of the current state s, taking action a under policy π.

$$Q_\pi(s,a) = \mathbb{E}_\pi[R_{t+1} + \gamma Q_\pi(S_{t+1}, A_{t+1}) | S_t = s, A_t = a]$$

The actor receives a state as input and outputs the probability of each action. This is policy-based learning that controls how the agent moves by learning the optimal policy. On the other hand, the critic evaluates the action by taking the state as input and calculating the value function (value-based learning). The two networks are trained separately, and the gradient ascent method is used to update the weights. As a result, the more the timestep is repeated, the better the actor will perform, and the better the critic will evaluate the actions.

In this study, the actor-critic method is used among several reinforcement learning techniques. The advantage of the actor-critic method is that the learning speed is fast because it learns every timestep. REINFORCE, a Monte-Carlo policy gradient method, and the learning speed is relatively slow because it learns for each episode.

Advantage actor-critic (A2C) is a method of using an advantage function. The

advantage function compares how good an action is compared to other actions in a given state. The reason for using the advantage function is that the larger the Q function value, the greater the variance of the error function. Therefore, to reduce the degree of change in the Q function, the value function, which is the baseline, is subtracted. The advantage function is as follows.

$$A(s_t, a_t) = Q(s_t, a_t) - V(s_t)$$

Table 4.3 Algorithm of actor-critic policy gradient

Algorithm: Action-Value Actor-Critic
function QAC
Initialize s, θ arbitrarily
Sample $a \sim \pi_\theta$
for each step **do**
Sample reward $r = R_s^a$; sample transition $s' \sim Pr_s^a$
Sample action $a' \sim \pi_\theta(s', a')$
Update policy parameters: $\theta = \theta + \alpha \nabla_\theta \log \pi_\theta(s_t, a_t)\, Q_w(s, a)$
Compute the correction for action-value at time t:
$$\delta = r + \gamma Q_w(s', a') - Q_w(s, a)$$
and use it to update action function parameters: $w \leftarrow \mathrm{w} + \beta \delta \nabla_w Q_w(s, a)$
Update $a \leftarrow a', s \leftarrow s'$
end for
end function

Chapter 5. Numerical Example

In this chapter, the developed model in Chapter 3 and Chapter 4 is applied to a real network. The description of data is referenced from Seo et al. (2020). The results are reported and compared. Historical usage data from 31 stations installed in Yeouido are used. As the input variable of the model changes, the relationship between the decision variables and the value of the objective function is identified.

5.1 Data Overview

5.1.1 Data Collection

Three datasets were used in this analysis: the bicycle sharing dataset, holiday data, and a meteorological dataset. The bicycle sharing system dataset was provided by the Seoul Facilities Corporation (SFC), a public bicycle management agency in Seoul. Holiday data were supplied by Open Data Source of South Korea, and the meteorological dataset was collected from the Korean Meteorological Administration (KMA).

The Seoul Bicycle Sharing (SBS) system can be used by applying for a pickup from a smartphone application or internet homepage. A bicycle is available for one or two hours and the types of members are either regular or casual members. The individual trip data include the following information:

- member type: whether the user was a regular or a casual member
- pickup time: pickup date and time
- pickup place: name and number of the pickup station
- return time: return date and time
- return place: name and number of the return station

In this study, an individual pickup record was aggregated on hourly basis, and the number of hourly pickups and returns were calculated. The aggregated period is labelled as year, season, month, day of the week, and time of day, and the labels are applied as input variables. Therefore, this study analyzes all time periods including weekdays and weekends. The number of hourly pickups and returns serve as a response variable.

Inventory data were collected every 10 minutes for each station. The inventory data include the following factors:

- station information: name and number of the station
- time: inventory collection time
- stock information: number of bicycles and capacity of the station

The Korea Astronomy and Space Science Institute provides holiday information through APIs on the website of an open data portal (see https://www.data.go.kr), which was utilized to determine public holiday information of South Korea from 2015 to 2017 was collected. These data were translated into a binary variable, one for holidays and zero otherwise. Meteorological data were collected from the website of the Korea Meteorological Administration (KMA), such as temperature, precipitation, and wind speed, for the same period as the bicycle pickup records. These data included hourly weather data from the Automatic Weather System (AWS) in a csv format.

5.1.2 Data preprocessing

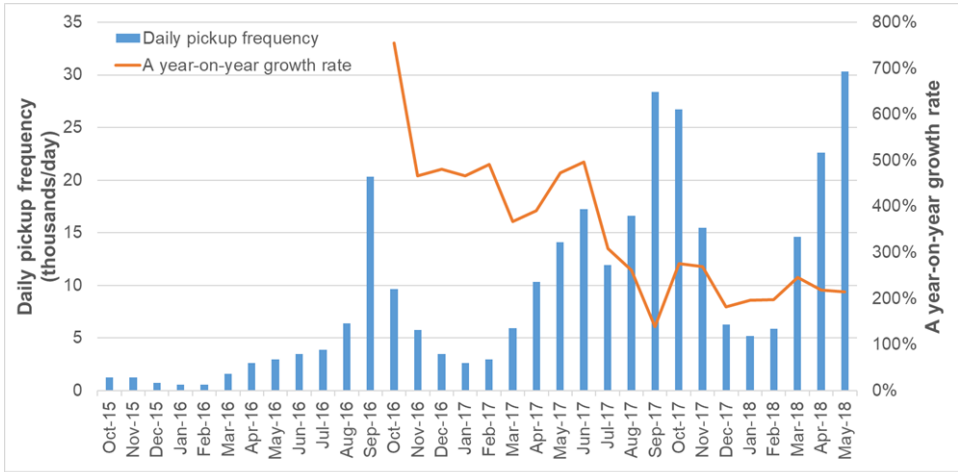
Due to the technical problems of the system or the loss or breakdown of a bicycle, usage data may be logged incorrectly. Therefore, it is necessary to preprocess data to forecast demand. Trip data of less than one minute or more than 24 hours of usage time were judged to be abnormal, and were removed. As a result, the study used 586,602 historical pickup data from January 1, 2016 to September 20, 2017.

Individual trip data were collected at intervals of one hour, with the number of pickups and returns. Temporal factors, such as year, season, month, day of the week, and hour were assigned to each time period. Meteorological factors, such as temperature, precipitation, and wind speed, were also combined with the corresponding time period.

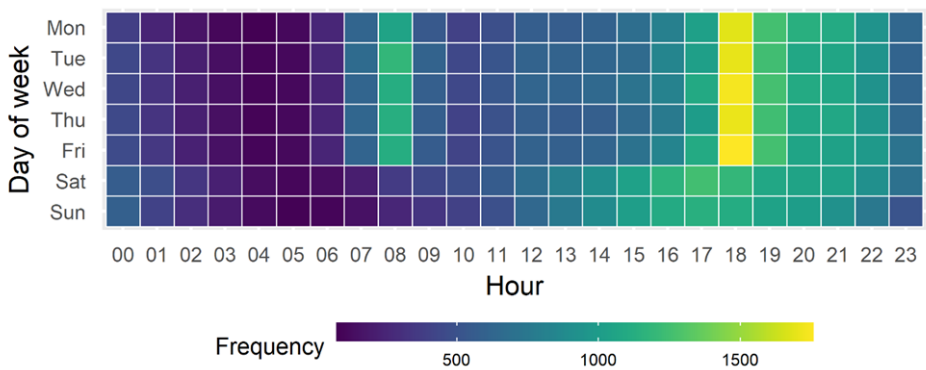
5.1.3 Descriptive statistics

Figure 5.1 (a) shows that usage has increased over the years since the system has been continuously expanded. During the entire period, there was an average of 9,000 pickups per day. In April 2018, there was an average of 22,826 pickups per day, which was 2.16 times increase from the same month one year earlier. Like the public bicycle systems in other cities (Rudloff and Lackner, 2014; Fournier et al., 2017), there are seasonal characteristics, such as much traffic from June to October in the summer and autumn and a decrease in traffic from December through February in the winter. These characteristics are the reason why the year, season, and month variables were added to the forecasting factors.

Figure 5.1 (b) shows the number of pickups by day and time of day as a heat map. On weekdays, the use of bicycles was expected to be high during the morning and evening because of commuting trips. On weekends, there were many pickups in the afternoon, which was assumed to be due to the use of bicycles for leisure activities. Regular members frequently used bicycles during morning peak hours and evening peak hours on weekdays, while casual members picked up more bicycles in the evening than in the morning. Therefore, day and time-of-day variables were considered as the demand prediction factors.



(a)



(b)

Figure 5.1 (a) The total number of pickups of bicycles, (b) daily pickup frequency heat map by day of the week and time of day

As shown in Figure 5.2, temperature, wind speed, and precipitation influenced public bicycle usage. The number of pickups had a positive correlation with a temperature of up to about 25 degrees Celsius and a negative correlation with a temperature over 25 degrees Celsius. Similar to the pattern of temperature, the pickup frequency tends to increase when the wind speed increases up to about 1.5 m/s, and to decrease when above 1.5 m/s. Meanwhile, rain had a significant negative impact on pickups even in small amounts. Because the number of operating bicycles in bad weather conditions decreases dramatically, weather factors should be included in the variables of demand forecasting.

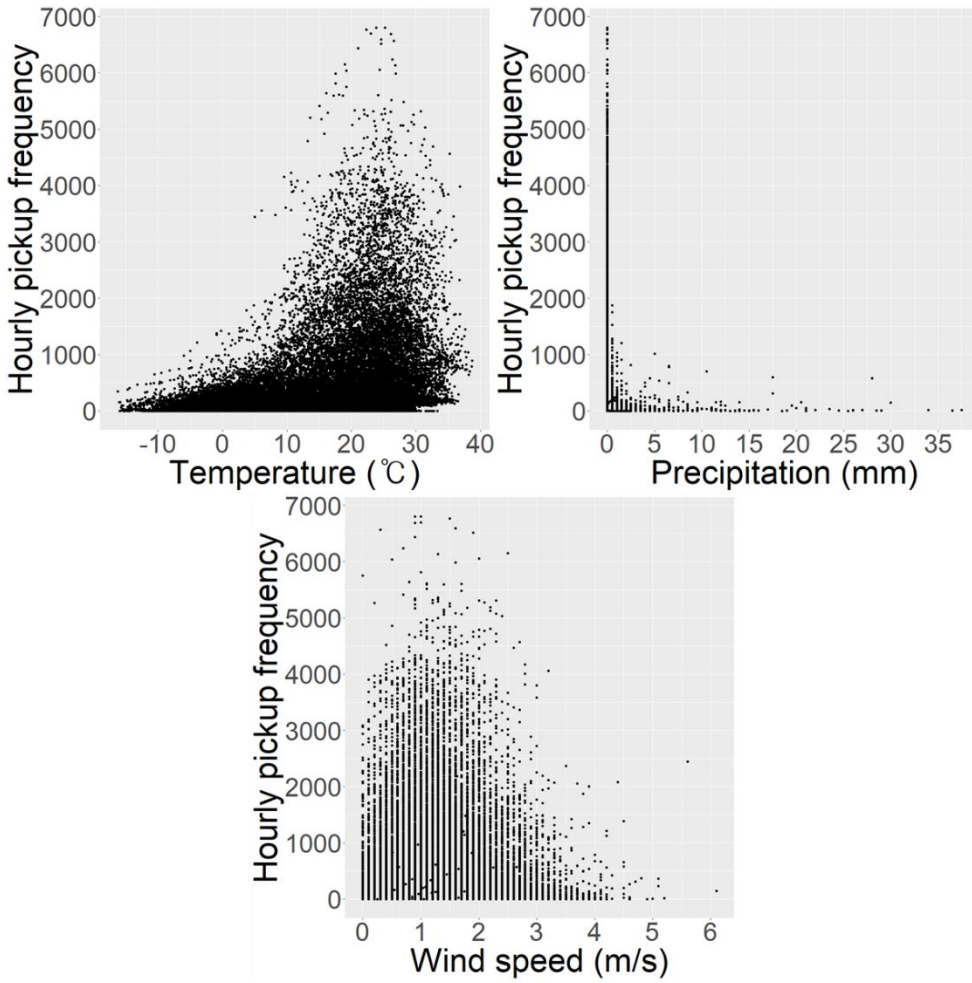


Figure 5.2 Relationship between meteorological factors and pickup frequency

Figure 5.3 shows the daily pickup frequency by month and time of the day during the analysis period of the PBS system in Seoul. Pickup frequency is high in fall and low in winter, and hourly pickup frequency stands out in the morning and afternoon peaks.

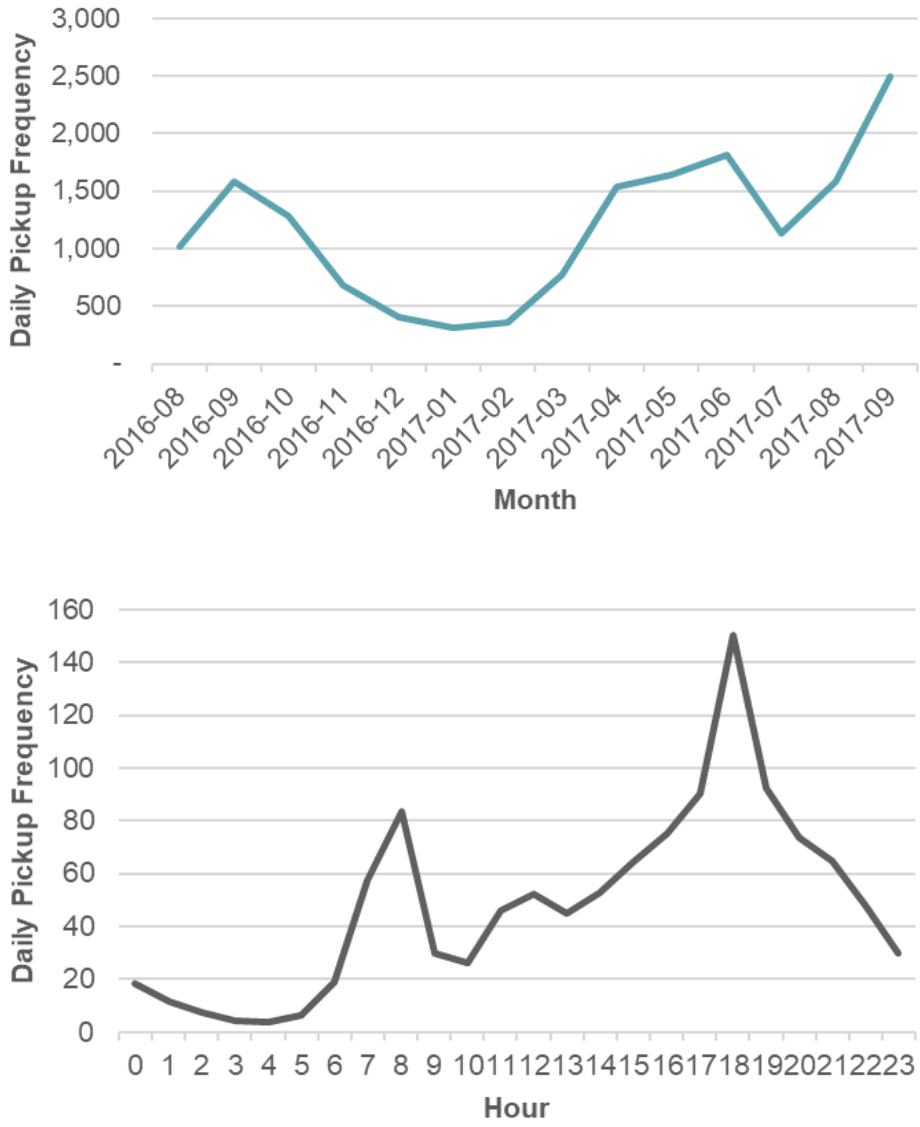


Figure 5.3 Daily pickup frequency by month (upper) and time of day (lower) of the PBS system in Seoul

5.2 Experimental Design

5.2.1 Key input data

5.2.1.2. Vehicle

Yeouido, the spatial scope of this study, has currently been relocated by one vehicle and the maximum number of bicycles that can be loaded on a vehicle is 15. This study also reflects this context, assuming that a vehicle with a capacity of 15 bicycles is responsible for the Yeouido area. It is assumed that the vehicle speed when moving between stations is 20 km/h, considering the average speed of traffic in Yeouido, and that it takes one minute per bicycle for withdrawing or distributing the bicycles.

5.2.1.3. Network

The network used in this chapter consists of 31 stations installed in Yeouido and a depot outside Yeouido. Based on the historical repositioning log, the unit timestep is set to 10 minutes. The Euclidean distance between stations is assumed, and the initial inventory at the start of the operation is assumed to be the observed inventory of the corresponding date. Table 5.1 shows the station-to-station travel time in Yeouido.

5.2.1.4. Demand

There are two types of demand used in this study: predicted demand and observed demand. To simulate the stochasticity of the demand, the observed demand is assumed to follow the Poisson distribution, and the forecasted demand is used as the mean of this distribution. Using historical data, the pickup and return demand in one hour is estimated at each station and uniformly assigned every 10 minutes, which is the unit of the time period. The time scope of the training set is set from August 2016,

the last time a station was installed in Yeouido, until the time just before the repositioning is carried out. The length of the prediction horizon is the next two hours.

The observed demand means the demand actually observed every 10 minutes at each station. In this chapter of this study, the observed demand is not used because the demand is a censored value and cannot reflect the potential demand. Therefore, it is assumed that the predicted demand occurs each timestep in the case study to compare strategies.

For convenience in understanding the results, it is necessary to identify the demand pattern in the analysis period. Three different demand patterns are shown in Figure 5.4. On a weekday morning, the number of returns was high, but since then, the number of pickups has kept higher than the number of returns. The number of pickups increases significantly during rush hour in case of the weekday evening (WE). Even this is presumed to have not shown all pickup demand due to a lack of inventory. On the weekend evening (WE), the number of pickups and returns was similar and more frequent than on weekdays.

5.2.2 Analysis conditions

In this study, three analysis periods are considered: weekday morning (07:00~13:00), weekday evening (16:00~22:00), and weekend evening (16:00~22:00). Demand patterns for analysis periods are shown in Figure 5.4. On weekdays morning, the number of returns has been more than the number of pickups, and on a weekday evening, the pickup has been more than the return. There are a lot of pickups and returns on a weekend evening. This usage pattern is due to the regional characteristics in Yeouido where several parks and dense office buildings are located. Since the effect of the strategy can be revealed in the period where there is a significant difference between pickup and return, the weekday morning was selected as analysis periods.

The length of the prediction horizon was set to two hours, considering the calculation time. Since the forecasted hourly demands were randomly assigned every 10 minutes at the demand forecasting module, the analysis was conducted five times, and the average values of KPIs were compared.

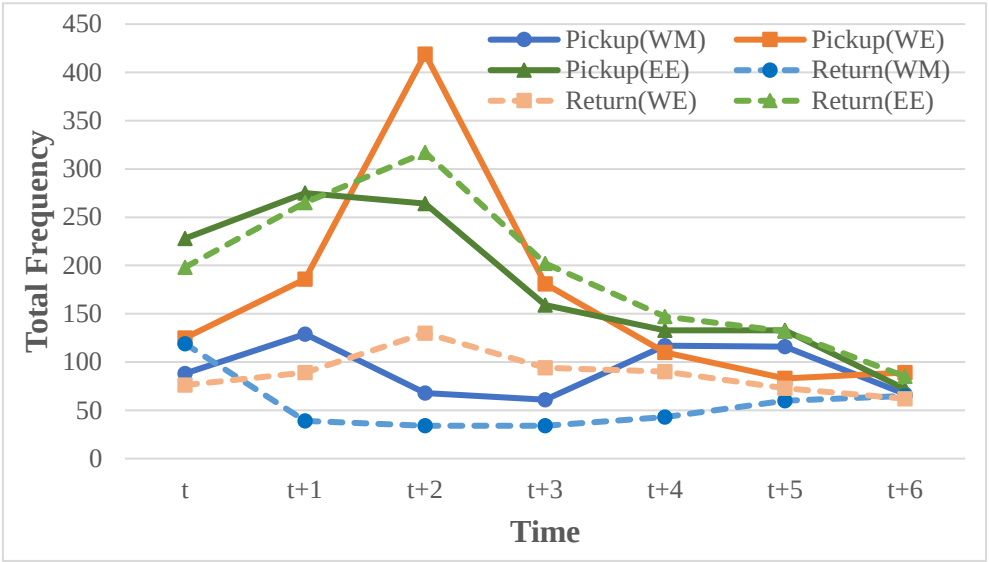


Figure 5.4 Demand patterns for analysis period

Table 5.1 Station-to-station travel time deployed in Yeouido

Station	Depot	ST-45	ST-46	ST-47	ST-51	ST-50	ST-52	ST-53	ST-73	ST-55	ST-56	ST-57	ST-58	ST-59	ST-60	ST-64	ST-65	ST-66	ST-61	ST-62	ST-63	ST-67	ST-68	ST-69	ST-70	ST-71	ST-72	ST-296	ST-297	ST-414	ST-424	ST-425
Depot	0	5	7	7	5	5	6	5	9	7	7	6	6	5	5	9	9	8	8	7	7	10	10	9	8	8	7	7	10	6	7	6
ST-45	5	0	2	3	2	1	2	2	5	4	3	4	4	3	3	6	6	5	5	4	5	7	7	6	6	6	6	3	6	2	4	4
ST-46	7	2	0	2	2	2	2	3	4	3	3	4	4	4	3	5	5	4	4	4	5	6	6	6	5	6	5	1	5	3	4	5
ST-47	7	3	2	0	2	3	2	3	3	2	1	3	3	3	3	4	3	3	3	3	4	5	5	4	4	5	4	1	4	4	3	4
ST-51	5	2	2	2	0	1	1	2	4	3	2	3	3	3	2	5	5	4	4	4	4	6	6	5	5	5	5	2	5	3	3	4
ST-50	5	1	2	3	1	0	2	2	5	4	3	3	3	3	2	5	5	5	4	4	4	6	7	6	5	5	5	3	6	3	4	4
ST-52	6	2	2	2	1	2	0	1	4	3	2	2	2	2	2	4	4	3	3	3	3	5	6	5	4	4	4	2	5	4	3	3
ST-53	5	2	3	3	2	2	1	0	4	3	3	2	2	1	1	5	4	4	4	3	3	6	6	5	5	4	4	3	5	4	3	3
ST-73	9	5	4	3	4	5	4	4	0	2	2	3	3	4	4	2	2	2	2	3	4	3	3	3	3	4	4	3	2	6	3	4
ST-55	7	4	3	2	3	4	3	3	2	0	1	2	2	3	2	2	2	2	1	2	2	3	4	3	2	3	3	3	3	6	1	3
ST-56	7	3	3	1	2	3	2	3	2	1	0	2	2	3	2	3	3	2	2	2	3	4	5	4	3	4	3	2	3	5	2	3
ST-57	6	4	4	3	3	3	2	2	3	2	2	0	1	2	2	3	3	2	2	1	2	4	4	3	3	3	2	3	4	6	1	2
ST-58	6	4	4	3	3	3	2	2	3	2	2	1	0	2	1	4	3	2	2	1	1	4	5	3	3	3	2	4	4	6	2	1
ST-59	5	3	4	3	3	3	2	1	4	3	3	2	2	0	1	5	4	4	3	2	3	6	6	4	4	4	3	4	5	5	3	2
ST-60	5	3	3	3	2	2	2	1	4	2	2	2	1	1	0	4	4	3	3	2	2	5	5	4	4	3	3	3	5	5	2	2

ST-64	9	6	5	4	5	5	4	5	2	2	3	3	4	5	4	0	1	2	2	3	3	2	2	2	2	3	3	4	1	7	3	5
ST-65	9	6	5	3	5	5	4	4	2	2	3	3	3	4	4	1	0	1	1	3	3	2	2	2	2	3	3	4	1	7	2	4
ST-66	8	5	4	3	4	5	3	4	2	2	2	2	2	4	3	2	1	0	1	2	2	2	3	2	1	2	2	4	2	7	1	3
ST-61	8	5	4	3	4	4	3	4	2	1	2	2	2	3	3	2	1	1	0	2	2	3	3	2	2	2	2	3	2	6	1	3
ST-62	7	4	4	3	4	4	3	3	3	2	2	1	1	2	2	3	3	2	2	0	1	4	4	2	2	2	2	4	4	6	1	2
ST-63	7	5	5	4	4	4	3	3	4	2	3	2	1	3	2	3	3	2	2	1	0	4	4	2	2	2	1	4	4	7	2	2
ST-67	10	7	6	5	6	6	5	6	3	3	4	4	4	6	5	2	2	2	3	4	4	0	1	2	2	3	3	5	1	8	3	5
ST-68	10	7	6	5	6	7	6	6	3	4	5	4	5	6	5	2	2	3	3	4	4	1	0	2	2	3	3	6	2	9	4	5
ST-69	9	6	6	4	5	6	5	5	3	3	4	3	3	4	4	2	2	2	2	2	2	2	2	2	0	1	2	2	5	2	8	2	4
ST-70	8	6	5	4	5	5	4	5	3	2	3	3	3	4	4	2	2	1	2	2	2	2	2	1	0	2	2	5	2	7	2	3
ST-71	8	6	6	5	5	5	4	4	4	3	4	3	3	4	3	3	3	2	2	2	2	3	3	2	2	0	1	5	3	8	2	3
ST-72	7	6	5	4	5	5	4	4	4	3	3	2	2	3	3	3	3	2	2	2	1	3	3	2	2	1	0	5	4	7	2	2
ST-296	7	3	1	1	2	3	2	3	3	3	2	3	4	4	3	4	4	4	3	4	4	5	6	5	5	5	5	0	5	4	3	5
ST-297	10	6	5	4	5	6	5	5	2	3	3	4	4	5	5	1	1	2	2	4	4	1	2	2	2	3	4	5	0	8	3	5
ST-414	6	2	3	4	3	3	4	4	6	6	5	6	6	5	5	7	7	7	6	6	7	8	9	8	7	8	7	4	8	0	6	6
ST-424	7	4	4	3	3	4	3	3	3	1	2	1	2	3	2	3	2	1	1	1	2	3	4	2	2	2	2	3	3	6	0	2
ST-425	6	4	5	4	4	4	3	3	4	3	3	2	1	2	2	5	4	3	3	2	2	5	5	4	3	3	2	5	5	6	2	0

Unit: mins

5.2.3 Details of computer, solver and programming environment

The details of computers and software used in this study are as follows. Solution algorithms are coded and compiled in the Python environment.

- Processor: AMD Ryzen 7 1700X Eight-Core Processor 3.40 GHz
- RAM: 40GB
- Operating system: Windows 10 Education 64-bit
- Python 3.6
    - PyCharm 2020.1 (Community Edition)
- R version: 4.0.0
    - Random forest: 'randomForest' package

5.3 Algorithm Performance

Numerical experiments with small-size problems are conducted first to compare the computational performance of the exact, approximate and reinforcement learning algorithms.

5.3.1 Network settings

For dynamic programming, four stations and the depot were selected randomly in Yeouido area. For RTDP, five to seven stations and the depot were selected. The number of timesteps is 12, which is over 2 hours from 6 p.m. to 8 p.m. on September 20, 2017. The discount factor γ is set to 0.9.

Dynamic programming was judged to have converged when the maximum change in a state value over a sweep was less than 10^{-1}. For RTDP, 1,000 iterations are performed.

5.3.2 Benchmark policies

Three benchmark strategies are modeled to assess the effectiveness of the developed strategies in this study. First two strategies were proposed by Brinkmann et al. (2015) and Brinkmann et al. (2019), and the third strategy is modeled based on the operations manual made by Seoul Facilities Corporation.

5.3.2.1. Short-term relocation policy

A short-term relocation (STR) policy was introduced by Brinkmann et al. (2015). Given a state (t_k, n_k^v, f_k^v, f_k), only one action is determined by the policy. If the fill levels of the current station are outside of the safety buffer, a relocation operation is implemented by the amount of the deficit (or the excess).

$$\iota^x = \begin{cases} \min\left\{\lceil \beta \cdot c(n_k^v) \rceil - f_k^{n_k^v}, f_k^v\right\}, & \text{if } f_k^{n_k^v} < \lceil \beta \cdot c(n_k^v) \rceil \\ \max\left\{\left\lceil (1-\beta) \cdot c(n_k^v) - f_k^{n_k^v} \right\rceil, f_k^v - c_v\right\}, & \text{if } \lceil (1-\beta) \cdot c(n_k^v) \rceil < f_k^{n_k^v} \end{cases}$$

If there are stations which violate the safety buffers, the agent chooses the nearest station (routing decision).

$$\sigma(n) = \begin{cases} \dfrac{1}{\tau(n_k^v, n)}, & \text{if } f_k^{n_k^v} < \lceil \beta \cdot c(n_k^v) \rceil \wedge 0 < f_k^v - \iota^x \\ \dfrac{1}{\tau(n_k^v, n)}, & \text{if } \lceil (1-\beta) \cdot c(n_k^v) \rceil < f_k^{n_k^v} \wedge f_k^v - \iota^x < c_v \\ 0, & \text{otherwise} \end{cases}$$

$$n^x = \arg\max_{n \in N} \sigma(n)$$

To compare with Seoul Facilities Corporation strategy to be discussed later

section, the safety buffer β is set to 0.2 in this analysis. In other words, the safety buffer is $[0.2, 0.8]$.

5.3.2.2. Static lookahead policy

Brinkmann et al. (2019) proposed both static and dynamic lookahead policies to solve the stochastic dynamic inventory routing problem in bicycle sharing systems. In this study, the static lookahead policy (SLA) is modeled. Determined inventory at the current station is the one of the three (25%, 50%, or 75% of station's capacity) leading to the least amount of unsatisfied demand as the sum of failed pickup and failed return demands at current station.

$$\gamma_{\iota,n}^{-} = \frac{1}{32}\sum_{j=1}^{32}\sum_{k^j=k}^{k_{\max}^j} p_j^{-}(s_{k^j}, n)$$

$$\gamma_{\iota,n}^{+} = \frac{1}{32}\sum_{j=1}^{32}\sum_{k^j=k}^{k_{\max}^j} p_j^{+}(s_{k^j}, n)$$

$$\iota^{\chi} = \operatorname{argmin}_{\iota\in\{\iota_1,\iota_2,\iota_3\}}\left\{\gamma_{\iota,n_k^v}^{-} + \gamma_{\iota,n_k^v}^{+}\right\}$$

where, $\gamma_{\iota,n}^{-}$, $\gamma_{\iota,n}^{+}$: the average number of failed pickups and returns for inventory decision ι and routing decision n

$p_j^{-}(\cdot,\cdot)$, $p_j^{+}(\cdot,\cdot)$: failed pickups and returns in simulation j

β: discount factor

At next decision, the agent selects the station n where relocations can prevent the largest amount of unsatisfied demand.

$$n^{\chi} = \arg\max_{n \in N} \{\min\{\gamma^{-}_{\iota^{\chi},n}, f^{t}_{v} - \iota^{\chi}\}, \min\{\gamma^{+}_{\iota^{\chi},n}, c_{v} - f^{t}_{v} - \iota^{\chi}\}\}$$

5.3.2.3. Seoul Facilities Corporation policy

According to the operation manual of SFC, the operating agency of the PBS system in Seoul, a repositioning employee tries to keep the inventory rate of the station between 20 percent and 80 percent. If the rate is violated, the employee moves to the station and carries out the repositioning. This policy is the same as STR policy in Section 5.3.2.1, except that the safety buffer is fixed as $\beta = 0.2$ and that the reposition amount is determined from the inventory, which is maintained at 50% of station capacity.

5.3.3 Comparison between dynamic programming and RTDP

Table 5.2 shows the performance comparison between dynamic programming and RTDP. Dynamic programming took several hours to compute to convergence even for small networks, but the calculation time of RTDP was drastically reduced. RTDP required only roughly a third of the updates that dynamic programming did. RTDP updated the values of 99.98% of the states no more than 100 times and 8,606 states were not updated at all in an average run. RTDP as well as reinforcement learning takes a lot of time to be applied in real-time. Still, it is more time-efficient than dynamic programming because RTDP and reinforcement learning can store values and update continuously.

Table 5.2 Performance comparison between dynamic programming and RTDP

	Dynamic programming	**RTDP**
Computation time	34,482 seconds	1,784 seconds
Average computation to convergence	3 sweeps	1,000 episodes
Average number of updates to convergence	26,400	8,578
Average number of updates per episode	-	8.6
% of states updated ≤ 100 times	-	99.98
% of states updated ≤ 10 times	-	99.24
% of states updated 0 times	-	81.50

5.3.4 Performances of RTDP

5.3.4.1. Computation time

The computation time of RTDP taken for ten iterations according to the number of stations is shown as Figure 5.5. The computation time by the number of stations grows exponentially with $|N|$. These observations are consistent with the computation of the dimension of the state space. As described in 3.2.1.2, each station has two values for a fill rate index (0 or 1), so the dimension of the state space is $|S| \leq |T| \cdot |N| \cdot 2^{|N|}$.

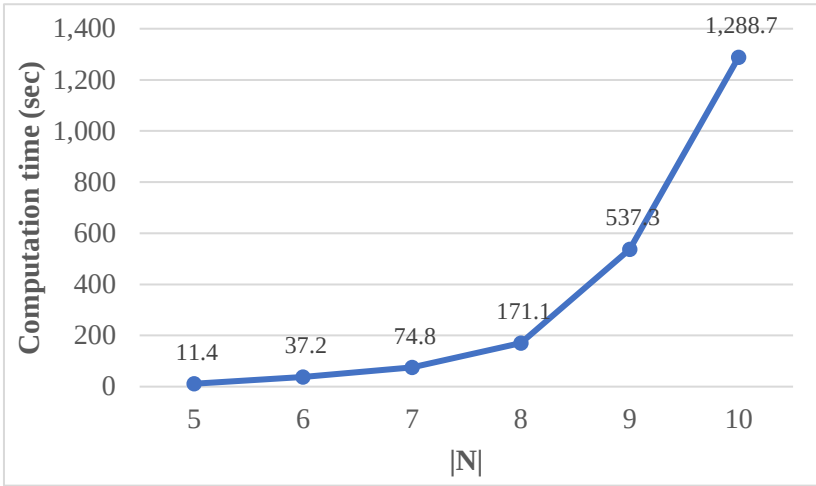


Figure 5.5 Computation time needed for 10 iterations

5.3.4.2. Comparison with benchmark policies

As illustrated in Section 5.3.2, three benchmark policies are analyzed to compare the effectiveness of the strategy developed in this study. Unmet demands, travel time, and delivery amount averaged from ten simulations for each policy. The unmet demands were considered for all stations and the delivery amount by the agent is the sum of both the number of loaded and unloaded bicycles. The results of the comparison are presented in Table 5.3 and Figure 5.6. For five stations and the depot, the analysis time is weekday morning from 07:00 to 09:00.

It is common for the delivery amount to decrease as vehicle travel time increases (or vice versa) within a limited time. However, when the idling of a repositioning vehicle is long or the movement due to the meaningless inventory decision $(\iota = 0)$ increases, both the delivery amount and the travel time decrease. SLA, a strategy that utilizes predicted demand information, has a lower performance than STR and SFC. The stations that need relocation are well selected in the SLA. However, they cannot be relocated due to restrictions on the inventory decision (at

least 25% of station capacity). As the analysis time was on the weekday morning, the demand for returns was higher than the demand for pickups. Even in a situation where all the bicycles at a station need to be withdrawn, the employee has no choice but to choose 25% of capacity. Besides, if revisit were allowed, the agent could have relocated bicycles later after the relocation staff person couldn't reposition them at the first visit.

Table 5.3 Key performance indicators by benchmark strategies

Strategies	**Average unmet demands**	**Average travel time (min)**	**Average delivery amount**
No reposition	10.6	-	-
STR(0.2)	8.5	19.8	2.7
SLA	9.6	40.2	11.7
SFC	5.8	19.8	5.7
RTDP(1.00)	3.8	37.1	16.8
RTDP(1.65)	3.5	37.2	21.9
RTDP(2.33)	2.3	38.3	21.4

STR is a strategy to visit as many stations as possible by relocating the minimum number of bicycles at the station. This strategy was found to be vulnerable to a sudden high demand. The failed demand occurred due to an intensive demand during the safety buffer or after the withdrawal of bicycles to prevent the station from filling up. Since STR and SFC are reactive strategies, the agent visits a station after a failure has already occurred. SLA has limited choices in inventory decisions (25%, 50%, or 75% of the station capacity) which make for unnecessary travel. In addition, the constraint that the agent can visit a station at most once makes the performance worse.

Overall, RTDP outperformed benchmark policies. In RTDP cases, the delivery amount was higher than other policies, and the delivery reduced the occurrence of

unmet demand. In particular, it had better performance when the z-score is high. More demand can be met by withdrawing bikes in the morning when return demand was intensive.

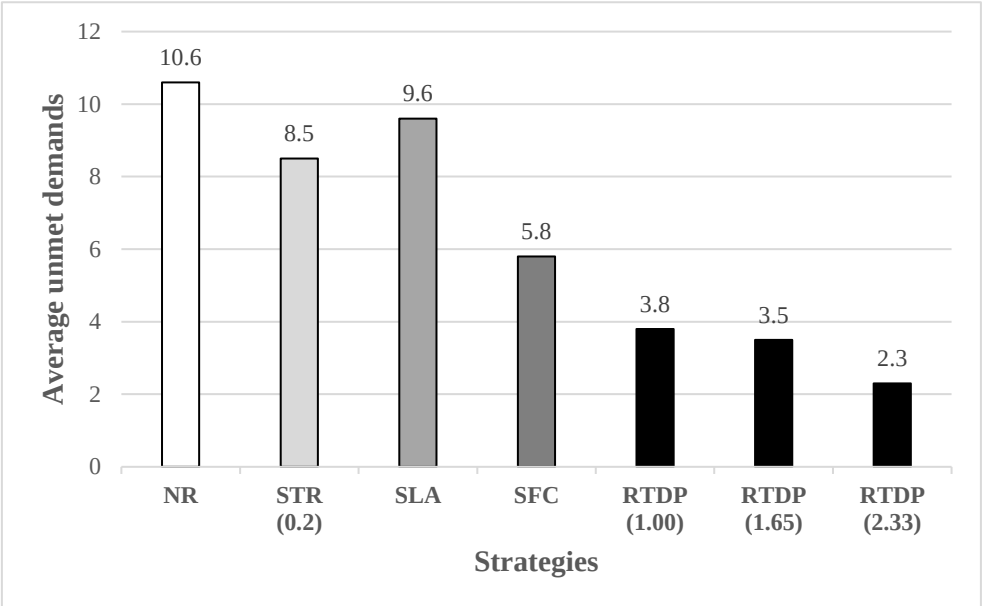


Figure 5.6 Comparison with benchmark policies

5.3.4.3. Comparison by strategies

As illustrated in Section 4.2.2, an agent considers three strategies:

- Strategy 1: Consider all stations
- Strategy 2: Consider stations close to the current station
- Strategy 3: Consider stations with large forecasting errors

For seven stations and depot, 100 iterations are performed by each strategy. The weight was set to 0.1 to account for the unmet demand and travel time together as a total cost. The cost of one unit of unmet demand was assumed to be 1,000 won, which is the price of a day voucher. Therefore, the cost of one minute of travel time for the agent is 100 won.

Table 5.4 shows KPIs for each strategy. A ‘Do nothing’ strategy means that no relocation is performed, and 22.8 average failures occurred in 2 hours. It is impossible to stay self-sufficient from only user usage because the pickup demand is higher than the return demand from 7 a.m. to 9 a.m., which is the analysis time. Therefore, this suggests that relocation using a truck is required.

All strategies showed better performance than a ‘Do nothing’ strategy. The rewards of Strategy 2 are similar to the value of a ‘Do nothing’ strategy, but the rewards of strategy 1 to 3 include the traveling cost of the vehicle. The reward of Strategy 1, considering all stations as the next to visit, was the lowest. In Strategy 3, the reward is similar to that of Strategy 1, but the computation time was reduced by about 28.5% compared to Strategy 1.

Table 5.4 Key performance indicators by strategies

Strategies	Total cost (won)	Time (s)
Do nothing: No reposition	22,800	-
Strategy 1: All stations	16,400	2,825.9
Strategy 2: Near stations	23,600	2,113.0
Strategy 3: Stations with large errors	16,800	2,021.6

5.4 Sensitivity Analysis

5.4.1 Z-score and safety buffer

This section describes the results of sensitivity analysis on safety stock and safety buffer. Increasing the safety stock means that an operator aims to prevent shortages with a high probability, so it makes sense to decrease the unmet demand as the Z-score increases. As shown in Table 5.5 and Figure 5.7, higher z can guarantee lower unmet demand. However, it is worth noting that higher cycle service levels require disproportionally higher Z-scores and disproportionately higher safety stock levels

(King, 2011). There is a trade-off between the number of stations to visit and the number of delivered bikes. In times of high demand, a higher z responds better to demand fluctuations.

A low safety buffer can cause the agent to serve a less urgent station, while a high safety buffer can make the employ serve a less critical station. A high safety buffer can lead to errors where an employee changes the priority of urgent stations. Therefore, it is useful to set an appropriate safety buffer, and this result is consistent with the results of the previous study, in which the performance curve was convex shape (Brinkmann et al., 2015; Brinkmann et al., 2019).

Table 5.5 Sensitivity analysis with varying Z-score and safety buffer

z \ β	**0.1**	**0.2**	**0.3**
1.00	5.8	3.8	4.0
1.65	5.0	3.5	4.0
2.33	4.7	2.3	2.8

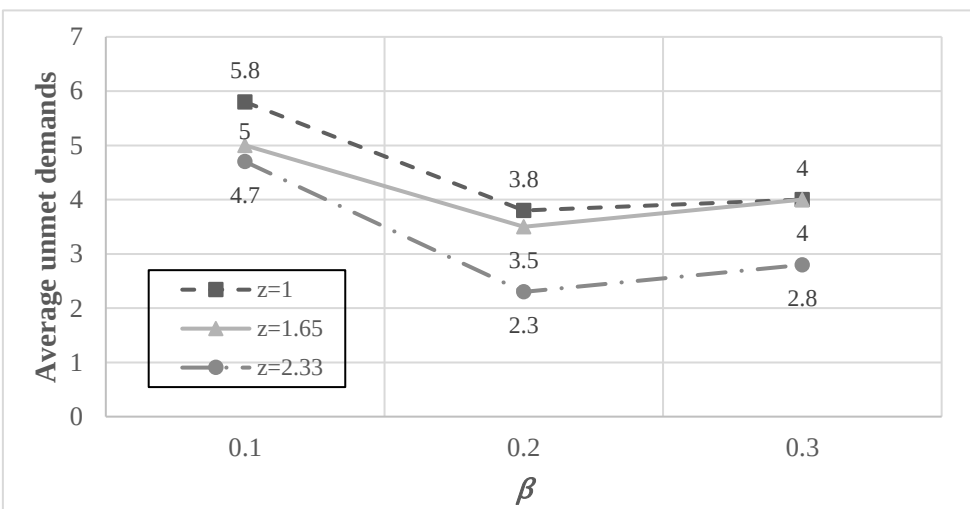


Figure 5.7 Sensitivity analysis with varying Z-score and safety buffer

5.5 Large-scale Cases

5.5.1 Network settings

All 31 stations and the depot were selected in Yeouido area for the large-scale case analysis ($|N| = 32$). An analysis time period is 20 Sep 2017, 07:00~09:00 (Weekday morning). Hyperparameters were set with reference to the values used in the literature. Hyperparameters in this analysis are as follows:

- Actor
    - Learning rate: 1×10^{-4}
    - Hidden layer units: 16
- Critic
    - Learning rate: 1×10^{-3}
    - Hidden layer units: 16

5.5.2 Results

5.5.2.1. Deterministic demand context

Figure 5.8 shows the performance analysis in a deterministic demand context. The deterministic demand used in this analysis is the observed demand on that day. Strategy 1, which searches all stations, requires more iterations to converge because the strategy takes more trial and error exploring all stations. Strategy 2 and 3, which search only a few stations, converge faster than Strategy 1 due to the reduction of the searching area. Among strategies, Strategy 3 has the lowest convergence.

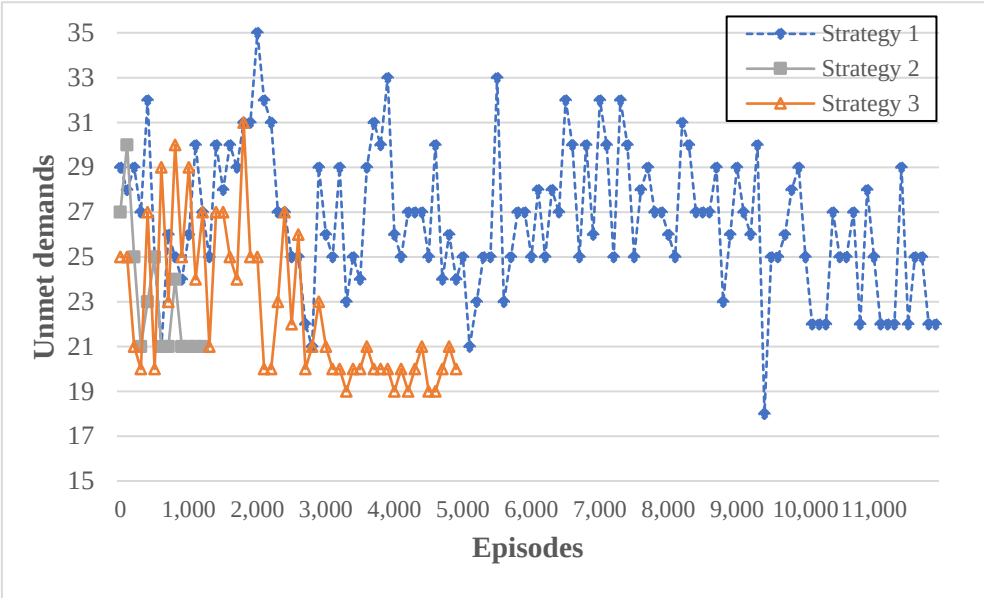


Figure 5.8 Performance analysis in deterministic demand context

Figure 5.9 to Figure 5.11 shows the repositioning results of each strategy. Inventory and routing decisions for each strategy are as follows:

- Strategy 1: Depot → ST-50 (+4) → ST-58 (+2) → ST-68 (+2) → ST-59 (+5) → ST-66 (+2) → ST-71 (-4) → Depot
- Strategy 2: Depot → ST-65 (+6) → ST-61 (+8) → ST-63 (-3) → ST-60 (+4) → Depot
- Strategy 3: Depot → ST-55 (+5) → ST-57 (+9) → Depot (-15) → ST-52 (+3) → ST-64 (+1) → Depot

An inefficient movement was observed in Strategy 1. Since all stations are candidates for routing decisions, the delivery amount is reduced by spending a lot of time on the move. It is impossible to serve all stations within a limited working time. In reality, repositioning staff people in the SBS system serve only about 20 stations for 9 hours (working time) due to handling broken bicycles or citizens' complaints. Therefore, a strategy is needed to select the station that needs the most relocation.

Strategy 2, which searches for nearby stations, can reduce travel time compared to Strategy 1, but cannot reduce total unmet demand by failing to serve distant

stations that need urgent relocation. This strategy may be useful for SBRP that aims to minimize total travel time, but it is not suitable for DBRP where pickup and return demand changes in real-time.

In Strategy 3, the agent goes through the depot again to withdraw bicycles for more delivery. On a weekday morning, most stations in Yeouido have much return demand rather than pickup demand due to commuting trips. As the analysis duration is short as 2 hours, the repositioning effect after the analysis period cannot be identified. If the analysis period becomes more extended, Strategy 3 will have better performance than other strategies.

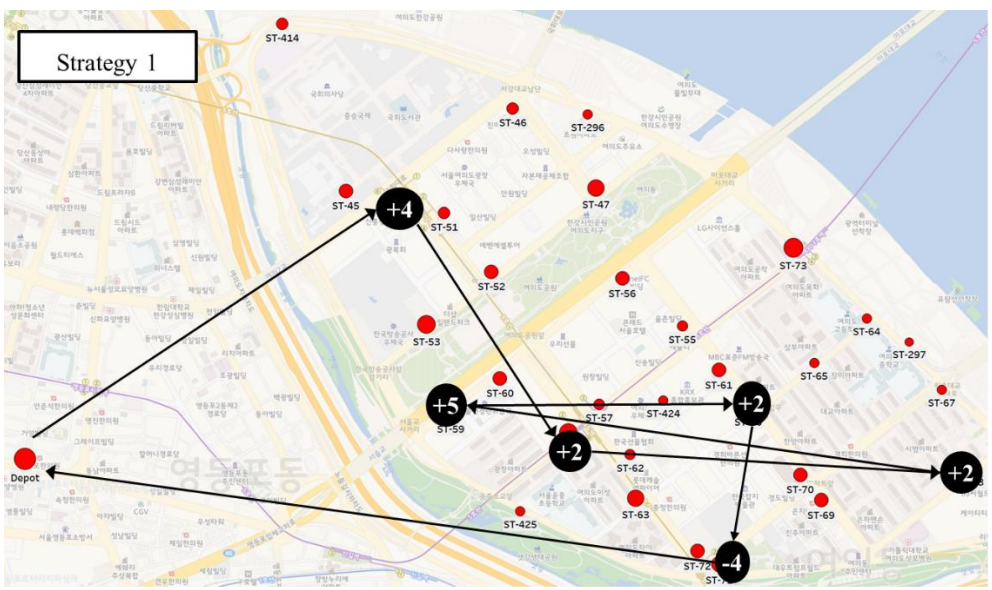


Figure 5.9 Repositioning result of Strategy 1

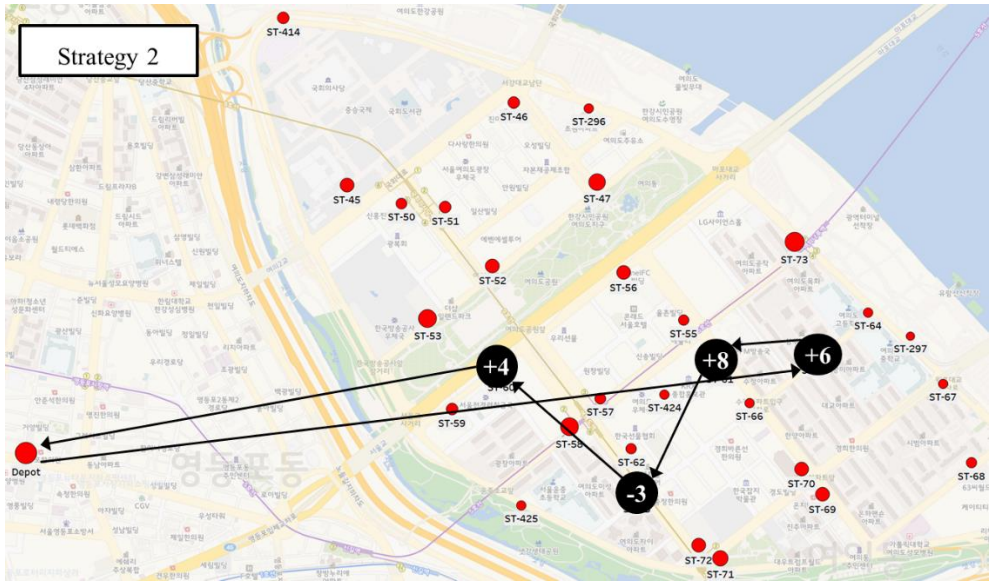


Figure 5.10 Repositioning result of Strategy 2

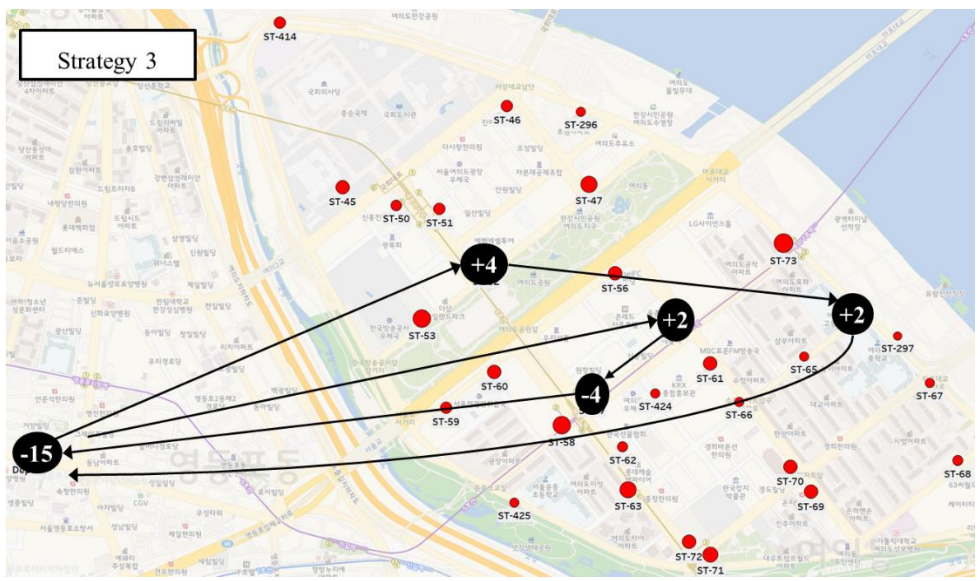


Figure 5.11 Repositioning result by Strategy 3

5.5.2.2. Stochastic demand context

In the stochastic demand context, the agent failed to minimize unmet demand with the current KPI. Because the unmet demand caused by the fluctuation of stochastic demand was larger than the unmet demand reduced by the agent's relocation, the

agent could not learn through the reward. In Figure 5.12, the variation in unmet demand is at most 50 bicycles. The agent, however, can only visit limited number of stations with a vehicle capacity of 15 bicycles. Since the unmet demand is much higher than the satisfied demand on weekday morning, it is necessary to increase the supply of public bicycles into Yeouido, including relocation with the repositioning vehicle.

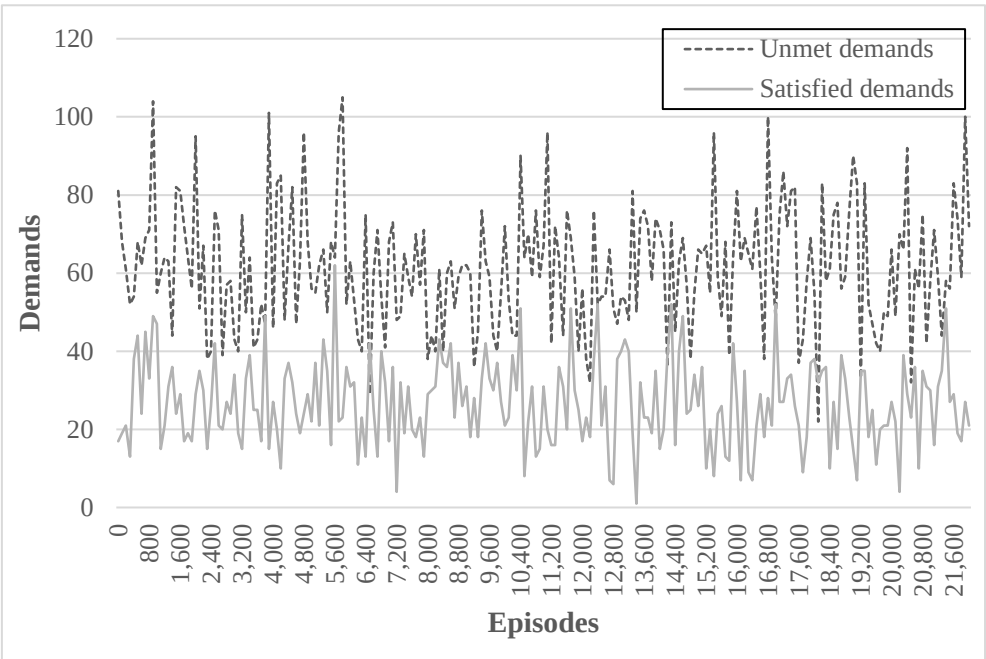


Figure 5.12 Performance analysis in stochastic demand context

This difficulty also applies if KPIs are changed into satisfied demands. Figure 5.12 shows that unmet demands and satisfied demands show the similar trend. Therefore, it is necessary to consider the reward by agent's action (inventory decision and routing decision), not the reward of the entire system. KPIs that focus on the agent's behavior are as follows:

- Number of repositioned bikes: The relocated bicycle must be assumed to be picked up and the vacant dock by the reposition must be assumed to be used for return. In addition, the agent may serve a station that can accommodate

a lot of bicycles due to a large capacity but that does not need a reposition.

- Satisfied demand of the repositioned bikes: The calculation of satisfied demands is complex and inaccurate because the demands can only be calculated for the corresponding timestep.
- Satisfied demands of the served station: The agent chooses a station where pickup or return demand is high regardless of the amount of relocation.
- Unmet demands of the served station: Contrary to the satisfied demands of the served station, the agent only finds stations with rare pickup or return demand, even the agent may choose not to travel.

Chapter 6. Conclusions

6.1 Conclusions

6.1.1 Summary

Many cities around the world have operated a PBS system to reduce air pollution and traffic congestion and to maintain citizens' health. Due to spatiotemporal demand patterns, however, a shortage of bicycles or docks inevitably occurs. Addressing the imbalance of bicycles is essential for the system to succeed, and accurate demand forecasting should also be implemented. Based on the forecasted demand, it is necessary to establish a repositioning strategy that tackles bicycle imbalance under stochastic demands.

This study developed dynamic programming, RTDP, and reinforcement learning methods in the context of the dynamic PBS system with stochastic demand. Analysis was done on user demand patterns which are different by time period based on historical pickup data. Demand forecasting was done stochastically with a random forest technique. The movement of vehicles over time was considered by introducing the MDP. The developed models and algorithms were compared with benchmark strategies, and the characteristics of the strategies were analyzed under various conditions as demand patterns and network characteristics were varied. Using the developed model and algorithms, we compared the proposed strategy and benchmark strategies under various conditions with demand patterns and network characteristics. The developed strategy resulted in better performance than the benchmark strategy. In other words, the strategy of focusing on stations with large fluctuations in dynamic factors showed a high repositioning effect.

Most of the previous research about the repositioning problem of public

bicycles dealt with the SBRP, in which the repositioning process was assumed to occur mainly at nighttime. Even though the publication of the DBRP studies has recently increased, users' demands have been regarded as constant or as deterministic values based on historical usage data. In this study, considering the stochastic forecasted demands, the vehicle route and the number of bicycles to load or unload were determined to minimize unmet demand. This study evaluated the repositioning strategy by considering the various dynamic factors. This study proposed a more efficient and reliable repositioning strategy and allows the selection of strategies to be applied under similar conditions when they are applied in the field.

6.1.2 Guidelines for repositioning

There are a couple of things to consider when applying this strategy to reality to add practical value. First, the network size that RTDP can calculate is relatively small. A strategic approach such as clustering methods for small regions is required to analyze a wide range of networks.

In order to apply this study to the field, the computation time of the algorithm should be reasonable. RTDP and reinforcement learning also take a lot of time to converge so it is difficult to apply the algorithms in the field. They are time-efficient because they can store values in advance and update continuously.

6.2 Future Research

6.2.1 Limitations

The observed demand was considered as a true demand in this study. Strictly speaking, the observed demand represents a low bound of demand, not the true demand. For example, if a station does not have any bicycle, it does not appear as an

observed demand, even if there is actually a pickup demand. Therefore, there is a limit in this study that the true pickup demand was underestimated, and in order to solve this problem, an estimate on the true demand is required. There are methods for estimating true demand using average using historical data only where all demand has been realized (O'Mahony, 2015), or simulations using historical data (Negahban, 2019).

Second, it takes a long time to apply the strategy. Since future repositioning strategies are derived by considering the dynamic factors of the latest time period (inventory, prediction error, or inventory rate variation), it is a long time to respond to rapidly changing demand. Considering the time, it takes to draw future demand and vehicle routes and the accuracy of demand forecast shorter than one hour (e.g., 10 minutes or 30 minutes), however, this time difference can be seen as inevitable and can be reduced by the development of computational skills and precise demand forecasting techniques.

Third, travel time between stations was considered static. In Yeouido, the travel time was reported to be similar regardless of the time period except for boulevards, so this assumption is judged to be reasonable within this study. However, if the spatial scope is expanded, the time-dependent travel time should be considered, and the actual travel time should be reflected using APIs rather than Euclidean distance.

6.2.2 Future research

This study assumed that the bicycle type was homogeneous. If the electric bicycle is introduced in the future, the type of public bicycles can be diversified and the repositioning strategy changes according to the users' characteristics to the electric bicycle. In addition, because the specifications of the electric bicycle are different from the existing bicycle, the combination of loading or unloading from/to the vehicle may be also various.

Further research is needed to increase the accuracy of public bicycle demand forecasting. There are two issues in demand forecasting for public bicycles: the first is the accuracy of forecasted demand itself and the second is the estimation of true demand (as described in Section 6.2.1). If demand forecasting accuracy is high, the movement of the vehicle is closer to the movement in the hindsight problem. Unmet demand can be more realistic by using estimated true demand rather than the observed demand.

This study tested repositioning strategies with limited resources in a relatively small spatial scope. Future research needs to consider the expansion of the range, and additional input of resources such as vehicles and staffs.

References

1. Adelman, D. (2004). A price-directed approach to stochastic inventory/routing. Operations Research, 52(4), pp. 499-514.
2. Alagoz, O., Hsu, H., Schaefer, A. J., and Roberts, M. S. (2010). Markov decision processes: a tool for sequential decision making under uncertainty. Medical Decision Making, 30(4), pp. 474-483.
3. Barth, M., and Todd, M. (1999). Simulation model performance analysis of a multiple station shared vehicle system. Transportation Research Part C: Emerging Technologies, 7(4), pp. 237–259. https://doi.org/10.1016/S0968-090X(99)00021-2
4. Barto, A. G., Bradtke, S. J., and Singh, S. P. (1995). Learning to act using real-time dynamic programming. Artificial intelligence, 72(1-2), pp. 81-138.
5. Bellman, R. (1957). Dynamic Programming. Princeton University Press.
6. Bertazzi, L., Bosco, A., Guerriero, F., and Lagana, D. (2013). A stochastic inventory routing problem with stock-out. Transportation Research Part C: Emerging Technologies, 27, pp. 89-107.
7. Braekers, K., Ramaekers, K., and Van Nieuwenhuyse, I. (2016). The vehicle routing problem: State of the art classification and review. Computers and Industrial Engineering, 99, pp. 300-313.
8. Breiman, L. (2001). Random forests. Machine learning, 45(1), pp. 5-32.
9. Breiman, L., Cutler, A., Liaw, A., and Wiener, M. (2018). Package 'randomForest'. https://cran.r-project.org/web/packages/randomForest/randomForest.pdf, Accessed 2020-July-24
10. Brinkmann, J., Ulmer, M. W., and Mattfeld, D. C. (2015). Short-term strategies for stochastic inventory routing in bike sharing systems. Transportation Research Procedia, 10, pp. 364-373.
11. Brinkmann, J., Ulmer, M. W., and Mattfeld, D. C. (2019). Dynamic lookahead policies for stochastic-dynamic inventory routing in bike sharing systems. Computers & Operations Research, 106, pp. 260-279.
12. Chemla, D., Meunier, F., and Wolfler-Calvo, R. (2011, March). Balancing a bike-sharing system with multiple vehicles. In Proceedings of Congress annual de la

société Française de recherche opérationelle et d'aidea la décision, ROADEF2011, Saint-Etienne, France.
13. Chiariotti, F., Pielli, C., Zanella, A., and Zorzi, M. (2018). A dynamic approach to rebalancing bike-sharing systems. Sensors, 18(2), p. 512.
14. Contardo, C., Morency, C., and Rousseau, L.-M. (2012). Balancing a dynamic public bike-sharing system. Cirrelt. Retrieved from https://www.cirrelt.ca/DocumentsTravail/CIRRELT-2012-09.pdf
15. Dantzig, G. B., and Ramser, J. H. (1959). The truck dispatching problem. Management science, 6(1), pp. 80-91.
16. Dell'Amico, M., Iori, M., Novellani, S., and Stützle, T. (2016). A destroy and repair algorithm for the bike sharing rebalancing problem. Computers & Operations Research, 71, pp. 149-162.
17. Erdoğan, G., Battarra, M., and Calvo, R. W. (2015). An exact algorithm for the static rebalancing problem arising in bicycle sharing systems. European Journal of Operational Research, 245(3), pp. 667-679.
18. Faghih-Imani, A. and Eluru, N. (2016). Incorporating the impact of spatio-temporal interactions on bicycle sharing system demand: A case study of New York CitiBike system. Journal of Transport Geography, 54, pp. 218-227.
19. Feng, Y., Wang, S. (2017), A forecast for bicycle rental demand based on random forests and multiple linear regression, In Computer and Information Science (ICIS), 2017 IEEE/ACIS 16th International Conference on (pp. 101-105). IEEE.
20. Fernández, A., Billhardt, H., Timón, S., Ruiz, C., Sánchez, Ó., and Bernabé, I. (2018, December). Balancing Strategies for Bike Sharing Systems. In International Conference on Agreement Technologies (pp. 208-222). Springer, Cham.
21. Fishman, E. (2016). Bikeshare: A review of recent literature. Transport Reviews, 36(1), pp. 92-113.
22. Fournier, N., Christofa, E., and Knodler Jr, M. A. (2017). A sinusoidal model for seasonal bicycle demand estimation. Transportation research part D: transport and environment, 50, pp. 154-169.
23. Froehlich, J. E., Neumann, J., and Oliver, N. (2009, June). Sensing and predicting the pulse of the city through shared bicycling. In Twenty-First International Joint Conference on Artificial Intelligence.
24. Hagen, K., and Gleditsch, M. D. (2018). A Column Generation Heuristic for the

Dynamic Rebalancing Problem in Bike Sharing Systems (Master's thesis). Norwegian University of Science and Technology.

25. Hastie, T., Tibshirani, R., and Friedman, J. (2009). The elements of statistical learning: data mining, inference, and prediction. Springer Science & Business Media.
26. Hernández-Pérez, H., and Salazar-González, J. J. (2004). A branch-and-cut algorithm for a traveling salesman problem with pickup and delivery. Discrete Applied Mathematics, 145(1), pp. 126–139. https://doi.org/10.1016/j.dam.2003.09.013
27. Hernández-Pérez, H., Salazar-González, J. J., and Santos-Hernández, B. (2018). Heuristic algorithm for the split-demand one-commodity pickup-and-delivery travelling salesman problem. Computers & Operations Research, 97, pp. 1-17.
28. Ho, S. C., and Szeto, W. Y. (2014). Solving a static repositioning problem in bike-sharing systems using iterated tabu search. Transportation Research Part E: Logistics and Transportation Review, 69, pp. 180-198.
29. Ho, S. C., and Szeto, W. Y. (2016). GRASP with path relinking for the selective pickup and delivery problem. Expert Systems with Applications, 51, pp. 14-25.
30. Ho, S. C., and Szeto, W. Y. (2017). A hybrid large neighborhood search for the static multi-vehicle bike-repositioning problem. Transportation Research Part B: Methodological, 95, pp. 340–363. https://doi.org/10.1016/j.trb.2016.11.003
31. Kang, S., Medina, J. C., and Ouyang, Y. (2008). Optimal operations of transportation fleet for unloading activities at container ports. Transportation Research Part B: Methodological, 42(10), pp. 970-984.
32. Kek, A. G., Cheu, R. L., and Chor, M. L. (2006). Relocation simulation model for multiple-station shared-use vehicle systems. Transportation research record, 1986(1), pp. 81-88.
33. Kek, A. G., Cheu, R. L., Meng, Q., and Fung, C. H. (2009). A decision support system for vehicle relocation operations in carsharing systems. Transportation Research Part E: Logistics and Transportation Review, 45(1), pp. 149-158.
34. King, P. L. (2011). Crack the code: Understanding safety stock and mastering its equations. APICS magazine, 21(2011), pp. 33-36.
35. Legros, B. (2019). Dynamic repositioning strategy in a bike-sharing system; how to prioritize and how to rebalance a bike station. European Journal of Operational Research, 272(2), pp. 740-753.

36. Lei, C., Ouyang, Y. (2018), Continuous approximation for demand balancing in solving large-scale one-commodity pickup and delivery problems, Transportation Research Part B: Methodological, 109, pp. 90-109.
37. Lessig, L. (2008). Remix: Making art and commerce thrive in the hybrid economy. Penguin.
38. Lin, J. R., and Yang, T. H. (2011). Strategic design of public bicycle sharing systems with service level constraints. Transportation research part E: logistics and transportation review, 47(2), pp. 284-294.
39. Lin, J. R., Yang, T. H., and Chang, Y. C. (2013). A hub location inventory model for bicycle sharing system design: Formulation and solution. Computers & Industrial Engineering, 65(1), pp. 77-86.
40. Lin, L., He, Z., and Peeta, S. (2018). Predicting station-level hourly demand in a large-scale bike-sharing network: A graph convolutional neural network approach. Transportation Research Part C: Emerging Technologies, 97, pp. 258-276.
41. Mahmoudi, M., and Zhou, X. (2016). Finding optimal solutions for vehicle routing problem with pickup and delivery services with time windows: A dynamic programming approach based on state-space-time network representations. Transportation Research Part B: Methodological, 89, pp. 19–42. https://doi.org/10.1016/j.ajic.2017.07.018
42. Meddin, R. (2018). The Bike-Sharing World map. www.bikesharingmap.com. (Accessed: 2018-June-01).
43. Mosheiov, G. (1994). The travelling salesman problem with pick-up and delivery. European Journal of Operational Research, 79(2), pp. 299-310.
44. Nath, R. B., and Rambha, T. (2019). Modelling Methods for Planning and Operation of Bike-Sharing Systems. Journal of the Indian Institute of Science, pp. 1-26.
45. Negahban, A. (2019). Simulation-based estimation of the real demand in bike-sharing systems in the presence of censoring. European Journal of Operational Research, 277(1), pp. 317-332.
46. Nourinejad, M., and Roorda, M. J. (2014). A dynamic carsharing decision support system. Transportation research part E: logistics and transportation review, 66, pp. 36-50.
47. O'Mahony, E. (2015). Smarter tools for (Citi) bike sharing. PhD thesis, Cornell

University.

48. Pan, L., Cai, Q., Fang, Z., Tang, P., and Huang, L. (2018). A Deep Reinforcement Learning Framework for Rebalancing Dockless Bike Sharing Systems. Retrieved from http://arxiv.org/abs/1802.04592
49. Parikh, P., and Ukkusuri, S. V. (2015). Estimation of Optimal Inventory Levels at Stations of a Bicycle-Sharing System (No. 15-5170).
50. Powell, W. B. (2011). Approximate Dynamic Programming: Solving the curses of dimensionality. John Wiley & Sons.
51. Puterman, M. L. (2014). Markov decision processes: discrete stochastic dynamic programming. John Wiley & Sons.
52. Raviv, T., Tzur, M., and Forma, I. A. (2013). Static repositioning in a bike-sharing system: models and solution approaches. EURO Journal on Transportation and Logistics, 2(3), pp. 187–229. https://doi.org/10.1007/s13676-012-0017-6
53. Regue, R., and Recker, W. (2014). Proactive vehicle routing with inferred demand to solve the bikesharing rebalancing problem. Transportation Research Part E: Logistics and Transportation Review, 72, pp. 192-209.
54. Repoux, M., Kaspi, M., Boyacı, B., and Geroliminis, N. (2019). Dynamic prediction-based relocation policies in one-way station-based carsharing systems with complete journey reservations. Transportation Research Part B: Methodological, 130, pp. 82-104.
55. Rixey, R. A. (2013). Station-level forecasting of bikesharing ridership: station network effects in three US systems. Transportation Research Record, 2387(1), pp. 46-55.
56. Rudloff, C., and Lackner, B. (2014). Modeling demand for bikesharing systems: neighboring stations as source for demand and reason for structural breaks. Transportation Research Record, 2430(1), pp. 1-11.
57. Schuijbroek, J., Hampshire, R. C., and Van Hoeve, W. J. (2017). Inventory rebalancing and vehicle routing in bike sharing systems. European Journal of Operational Research, 257(3), pp. 992-1004.
58. Seo, Y. H., Yoon, S., Kim, D. K., Kho, S. Y., and Hwang, J. (2020). Predicting Demand for a Bike-Sharing System with Station Activity Based on Random Forest. Proceedings of the Institution of Civil Engineers - Municipal Engineer. https://doi.org/10.1680/jmuen.20.00001

59. Shaheen, S. A., Guzman, S., and Zhang, H. (2010). Bikesharing in Europe, the Americas, and Asia: past, present, and future. Transportation Research Record, 2143(1), pp. 159-167.
60. Shui, C. S., and Szeto, W. Y. (2018). Dynamic green bike repositioning problem – A hybrid rolling horizon artificial bee colony algorithm approach. Transportation Research Part D: Transport and Environment. https://doi.org/10.1016/j.trd.2017.06.023
61. Singhvi, D., Singhvi, S., Frazier, P. I., Henderson, S. G., O'Mahony, E., Shmoys, D. B., and Woodard, D. B. (2015, April). Predicting bike usage for new york city's bike sharing system. In Workshops at the Twenty-Ninth AAAI Conference on Artificial Intelligence.
62. Shin, H., Kim, D., and Jeong, S. (2012). Impact Analysis on Bike-Sharing and Its Improvement Plan. The Korea Transport Institute. (Korean)
63. Sutton, R. S. and Barto, A. G. (2018). Reinforcement learning: An introduction. MIT press.
64. Szeto, W. Y., Liu, Y., and Ho, S. C. (2016). Chemical reaction optimization for solving a static bike repositioning problem. Transportation Research Part D, 47, pp. 104–135. https://doi.org/10.1016/j.trd.2016.05.005
65. Tang, Q., Fu, Z., and Qiu, M. (2019). A bilevel programming model and algorithm for the static bike repositioning problem. Journal of Advanced Transportation, 2019.
66. Wang, T. (2014). Solving dynamic repositioning problem for bicycle sharing systems: model, heuristics, and decomposition. Master thesis, The University of Texas at Austin.
67. Yang, Z., Hu, J., Shu, Y., Cheng, P., Chen, J., and Moscibroda, T. (2016, June). Mobility modeling and prediction in bike-sharing systems. In Proceedings of the 14th annual international conference on mobile systems, applications, and services (pp. 165-178). ACM.
68. Zhang, D., Yu, C., Desai, J., Lau, H. Y. K., Srivathsan, S. (2017), A time-space network flow approach to dynamic repositioning in bicycle sharing systems, Transportation research part B: methodological, 103, pp. 188-207.

초 록

실시간 동적 계획법 및 강화학습 기반의 공공자전거 시스템의 동적 재배치 전략

서울대학교 대학원
공과대학 건설환경공학부
서 영 현

공공자전거 시스템은 교통혼잡과 대기오염 등 여러 도시문제를 완화할 수 있는 교통수단이다. 대여소가 위치한 곳이면 언제 어디서든 이용자가 자전거를 이용할 수 있는 시스템의 특성상 수요의 시공간적 불균형으로 인해 대여 실패 또는 반납 실패가 발생한다. 시스템 실패를 예방하기 위해 운영자는 적절한 재배치 전략을 수립해야 한다. 운영자는 예측 수요 정보를 전제로 의사결정을 하므로 수요예측의 정확성이 중요한 요소이나, 수요의 불확실성으로 인해 예측 오차의 발생이 불가피하다.

본 연구의 목적은 공공자전거 수요의 불확실성과 시스템의 동적 특성을 고려하여 불만족 수요를 최소화하는 재배치 모형을 개발하는 것이다. 공공자전거 재배치 메커니즘은 순차적 의사결정 문제에 해당하므로, 본 연구에서는 순차적 의사결정 문제를 모형화할 수 있는 마르코프 결정 과정을 적용한다. 마르코프 결정 과정을 풀기 위해 복잡한 문제를 간단한 부문제로 분해하여 정확해를 도출하는 동적 계획법을 이용한다. 하지

만 마르코프 결정 과정의 상태 집합과 결정 집합의 크기가 커지면 계산 복잡도가 증가하므로, 동적 계획법을 이용한 정확해를 도출할 수 없다. 이를 해결하기 위해 근사적 동적 계획법을 도입하여 근사해를 도출하며, 대규모 공공자전거 네트워크에서 가능해를 얻기 위해 강화학습 모형을 적용한다. 장래 공공자전거 이용수요의 불확실성을 모사하기 위해, 기계학습 기법의 일종인 random forest로 예측 수요를 도출하고, 예측 수요를 평균으로 하는 포아송 분포를 따라 수요를 확률적으로 발생시켰다.

본 연구에서는 관측 수요와 예측 수요 간의 차이인 예측오차에 빠르게 대응하는 재배치 전략을 개발하고 효과를 평가한다. 개발된 전략의 우수성을 검증하기 위해, 기존 연구의 재배치 전략 및 현실에서 적용되는 전략을 모형화하고 결과를 비교한다. 또한, 재고량의 안전 구간 및 안전재고량에 관한 민감도 분석을 수행하여 함의점을 제시한다.

개발된 전략의 효과를 분석한 결과, 기존 연구의 전략 및 현실에서 적용되는 전략보다 개선된 성능을 보이며, 특히 예측오차가 큰 대여소를 탐색하는 전략이 전체 대여소를 탐색하는 전략과 재배치 효과가 유사하면서도 계산시간을 절감할 수 있는 것으로 나타났다. 공공자전거 인프라를 확대하지 않고도 운영의 효율화를 통해 공공자전거 시스템의 이용률 및 신뢰성을 제고할 수 있고, 공공자전거 재배치에 관한 정책적 함의점을 제시한다는 점에서 본 연구의 의의가 있다.

주요어 : 강화학습, 공공자전거 시스템, 마르코프 결정 과정, 실시간 동적 계획법, 재배치

학 번 : 2014-21505