Linear Control Policies for Online Vehicle Relocation in Shared Mobility Systems

Yasuhiro Yoshida[a], Yuichi Takano[b,*]

[a]College of Policy and Planning Sciences, University of Tsukuba, 1-1-1
Tennodai, Tsukuba-shi, 305-8573, Ibaraki, Japan
[b]Faculty of Engineering, Information and Systems, University of Tsukuba, 1-1-1
Tennodai, Tsukuba-shi, 305-8573, Ibaraki, Japan

Abstract

In one-way station-based shared mobility systems, where system users share vehicles for making trips between vehicle stations, the accumulation of one-way trips inevitably causes vehicle imbalances between stations. To correct these imbalances, we focus on the effective use of linear control policies for calculating online vehicle relocations from a history of user trips. Our scenario-based optimization model for computing linear control policies is formulated as a linear optimization problem. Computational results using a real-world dataset demonstrate that our method provides high relocation performance with short online computation times.

Keywords: shared mobility, vehicle relocation, control policy, optimization model

1. Introduction

Shared mobility systems allow for shared use of vehicles (e.g., cars and bicycles), which enables users to have short-term access to these transportation modes on an as-needed basis. Worldwide, there were approximately 15 million carsharing members in October 2016, and over 1600 public bikesharing systems based on information technology with over 18.17 million bicycles as of May 2018 [38]. Carsharing users can gain the benefits of private car use without the costs and responsibilities of ownership. Carsharing systems also promote greater use of alternative eco-friendly modes of transportation

[*] Corresponding author

(e.g., buses, bicycles, and walking) and thus are expected to yield considerable savings in the resources required to manufacture, maintain, operate, and store vehicles [25].

In one-way station-based shared mobility systems (e.g., Bay Wheels[1] and Ha:mo RIDE[2]), a user requiring a vehicle picks it up from a station (an origin station), makes a trip using the vehicle, and then drops it off at another station (a destination station). In contrast to two-way (or roundtrip) systems, one-way systems provide greater flexibility for users and make it possible to further enhance first- and last-mile connectivity [38]. However, the accumulation of one-way trips inevitably causes vehicle imbalances between stations. This is expected to result in empty or full stations, thereby degrading service levels and customer satisfaction.

To correct such vehicle imbalances, we consider the problem of relocating vehicles between stations in shared mobility systems. Although vehicles can be relocated partly by providing price incentives to users [23, 35, 45], a more practical method is staff-based relocation, in which staff drives vehicles or uses trucks to move vehicles to the desired stations. Staff-based relocation can be classified into static and dynamic methods [1]. Static relocation is conducted when the system is closed usually during the night, whereas dynamic relocation is more flexible and implemented throughout the day. There are offline and online methods for staff-based dynamic relocation [1]. In offline methods, the relocation plan to be executed throughout the day is determined at the beginning of the day. In online methods, relocation plans are dynamically modified in response to the changing situation during the day.

The various approaches to dealing with the vehicle relocation problem can be categorized into optimization models, simulation models, and multistage approaches [21]; see systematic reviews [1, 15, 18, 21, 28, 32] for detailed lists of research papers about the vehicle relocation problem.

In approaches based on optimization models, the vehicle relocation problem is formulated as mixed-integer optimization (MIO) models [7, 8, 19, 31, 34, 44]. Although these MIO models are particularly suitable for offline relocation and perform overall optimization of the system, the size of manageable MIO models is limited. In contrast, simulation models allow the stochastic

[1] `https://www.lyft.com/bikes/bay-wheels`

[2] `https://global.toyota/en/mobility/`

behavior of large-scale systems to be closely analyzed [3, 9, 20, 27, 37]. Although relocation performance is assessed through repeated simulation runs, the optimality of a selected relocation strategy cannot be guaranteed [22].

The advantages of optimization and simulation models are combined in multistage approaches [24, 26, 33, 36], where vehicle relocations are calculated by optimization models through the course of a simulation process. Recently, Calafiore et al. [12, 13] proposed online relocation strategies based on model predictive control (MPC) [14] for one-way station-based shared mobility systems. In each time period, this MPC method predicts user trips in future periods, calculates the sequence of optimal relocations, and executes the optimal relocation for the next period. The relocation performance of this method has been validated on a real-world dataset, but the repeated calculation of optimal relocations in each time period is computationally intensive.

To reduce the computational load of such multistage approaches, we consider designing control policies (i.e., decision rules) for computing appropriate relocations based on the current situation. In general, however, selecting the best control policy from among nonlinear functions is computationally intractable. Accordingly, we focus on the effective use of linear control policies, namely, decision rules restricted to the class of affine functions. Although this restriction may cause a substantial loss of optimality [5, 17], linear control policies have lower computational complexity for dynamic decision-making problems under uncertainty [4, 39]. Indeed, linear control policies have been used effectively for dynamic portfolio selection to produce low-risk high-return financial investments [10, 11, 40, 43]. Yet, to our knowledge, no prior studies have applied linear control policies to the vehicle relocation problem.

We propose a scenario-based optimization model for computing effective linear control policies for online vehicle relocation. In this model, the one-way station-based shared mobility system is represented as a time-space network. Our optimization model is formulated as a linear optimization problem, which can be solved to optimality by using optimization software.

The effectiveness of our method is assessed through computational experiments using a real-world dataset from the SF Bay Area Bike Share. The computational results demonstrate that our linear control policies are comparable to MPC-based relocation strategies [12, 13] in terms of relocation performance. Moreover, our method was significantly faster than MPC methods at online computation for relocating vehicles. To further validate our method, we also analyze the occupancy state of each station and relocation strategies that result from our linear control policies.

2. Optimization Model

This section first describes the vehicle relocation problem being addressed, and then presents our optimization model based on a time-space network for computing linear control policies.

2.1. Problem Description

Shared mobility systems provide users with shared vehicles (e.g., cars and bicycles), which are parked at vehicle stations in the neighborhoods of offices, bus stops, train stations, and tourist spots. A user of the system gets on a vehicle at a station of their choice. The procedure for starting to use the vehicle is generally executed via a smartphone app or system website. The user then drives the vehicle to their destination and parks it in a nearby vehicle station. The vehicle use is generally terminated by paying fares through a smartphone app or system website. Each one-way short-distance movement by a shared vehicle is called a *trip*, and users generally take a direct route between different stations as in the case of other transportation modes (e.g, taxi and bus).

A large number of one-way trips will inevitably cause vehicle imbalances between stations in shared mobility systems. For example, stations near major train stations tend to run out of vehicles during the morning commute, and those near popular tourist spots are likely to become full. Such vehicle imbalances increase user dissatisfaction, thereby leading to a significant loss of business from a long-term perspective. To reduce vehicle imbalances between stations, system managers need to relocate vehicles based on an analysis of vehicle usage history. We particularly focus on the situation where vehicles are relocated by staff from one station to another during operating hours.

Our optimization model is based on a mathematical model [12, 13] in which the actual complex systems are adequately simplified to develop high-performance MPC methods. In reality, each station has a limited capacity of parking spaces. Consequently, when the minimum/maximum number of vehicles at a station is reached, trip demands from/to the station cannot be approved. However, accurate modeling of this situation worsens the tractability of associated optimization problems. For this reason, we assume that all trip demands are fulfilled regardless of the number of vehicles at each station. As a result, we permit the possibility of violating the station capacity (i.e., minimum/maximum number of vehicles), and at the same time, such capacity violations are minimized in the optimization problem.

Throughout this paper, we also make the following assumptions about user trips and staff relocations.

1. Pre-booking of vehicle use and parking space is unnecessary.
2. The trip and relocation times include the time to prepare for movement and that to get on and off a vehicle.
3. Multiple scenarios are available, where each scenario is a history of user trips during one day.
4. Labor and fuel costs of relocation are ignored.
5. Movement of staff and trucks for relocation is not considered.
6. The relocation effort is quantified by the number of relocated vehicles.
7. The state of charge of batteries in electric vehicles is ignored.

2.2. Notation

We use the following notation in this paper.

Index Sets

D : Vehicle stations

S : Scenarios of a series of user trips in one day

$T := \{1, 2, \dots, t_{\mathrm{end}}\}$: Time periods during one day

T_{rel} : Time periods for launching relocation ($T_{\mathrm{rel}} \subseteq T$)

Parameters

$C_j^{\min}, C_j^{\max}$: Minimum/maximum number of vehicles (i.e., capacity) at station j

$N_{ij,s}^{\mathrm{dep}}(t)$: Number of trips departing from station i for station j in time period t under scenario s

$N_{ij,s}^{\mathrm{arr}}(t)$: Number of trips arriving at station j from station i in time period t under scenario s

R : Maximum number of vehicles to relocate in each time period

Z : Total number of vehicles in the system

α, β, γ : Weights of three objectives

τ_{ij} : Number of time periods required to relocate vehicles from station i to station j

τ_{pol} : Number of input time periods for control policies

Decision Variables

$a_{j,s}(t)$: Capacity violation of station j at the beginning of time period t under scenario s

$r_{ij,s}(t)$: Number of vehicles to relocate from station i to station j in time period t under scenario s

$r_{ij}(t)$: Net number of vehicles to relocate from station i to station j in time period t

$w_{ij}(t)$: Input weight of a control policy for relocating vehicles from station i to station j in time period t

$y^+_{j,s}, y^-_{j,s}$: Deviations from the initial condition at station j under scenario s

z^{ini}_j : Initial number of vehicles at station j

$z_{j,s}(t)$: Number of vehicles at station j at the beginning of time period t under scenario s

2.3. Time-space Network

The shared mobility system is represented as a *time-space network*, over which vehicles move due to trips by users and relocations by staff depending on each scenario $s \in S$. Each node in the time-space network is denoted by $(j, t) \in D \times T$, where D and T are the index sets of vehicle stations and time periods, respectively. The number of vehicles at node $(j, t) \in D \times T$ is $z_{j,s}(t)$ in each scenario $s \in S$.

There are Z vehicles available in the shared mobility system. The initial number of vehicles (z^{ini}_j) at station $j \in D$ must therefore satisfy the following constraints:

$$\sum_{j \in D} z^{\mathrm{ini}}_j = Z, \tag{1}$$

$$z^{\mathrm{ini}}_j \geq 0. \tag{2}$$

Each scenario $s \in S$ corresponds to a collection of trips made by system users during a single day, where $N^{\mathrm{dep}}_{ij,s}(t)$ and $N^{\mathrm{arr}}_{ij,s}(t)$ are the numbers of departure and arrival trips for station pair $(i, j) \in D \times D$ in period $t \in T$. We relocate $r_{ij,s}(t)$ vehicles from station $i \in D$ to station $j \in D$ in period $t \in T$ under scenario $s \in S$. Note that $r_{ij,s}(t) = 0$ for $t \leq 0$, and that the relocation requires τ_{ij} periods.

In each scenario $s \in S$, the number of vehicles ($z_{j,s}(t)$) at station $j \in D$ changes during period $t \in T$ according to the following *flow conservation* *constraint*:

$$z_{j,s}(t+1) = z_{j,s}(t) \underbrace{- \sum_{h \in D} N^{\mathrm{dep}}_{jh,s}(t) - \sum_{h \in D} r_{jh,s}(t)}_{\text{outflow by trips and relocations}} \underbrace{+ \sum_{h \in D} N^{\mathrm{arr}}_{hj,s}(t) + \sum_{h \in D} r_{hj,s}(t - \tau_{hj})}_{\text{inflow by trips and relocations}} \tag{3}$$

with the initial condition

$$z_{j,s}(1) = z^{\mathrm{ini}}_j. \tag{4}$$

2.4. Control Policies

To define control policies for online vehicle relocation, we first consider input information available at the beginning of each time period. Let us denote the past states of vehicle stations by $\boldsymbol{z}_s(t)$ and the history of user trips by $\boldsymbol{N}_s(t)$. These are formally defined for each scenario $s \in S$ and period $t \in T$ as follows:

$$\begin{aligned} \boldsymbol{z}_s(t) &:= (z_{j,s}(k) \mid j \in D, 1 \le k \le t), \\ \boldsymbol{N}_s(t) &:= ((N^{\mathrm{dep}}_{ij,s}(k), N^{\mathrm{arr}}_{ij,s}(k)) \mid i, j \in D, 1 \le k \le t-1). \end{aligned}$$

We then introduce a control policy $F_{ij,t}$, which outputs the net number of relocations (i.e., $r_{ij,s}(t) - r_{ji,s}(t)$) from the available information at the beginning of period $t \in T$. This relationship is given by

$$r_{ij,s}(t) - r_{ji,s}(t) = F_{ij,t}(\boldsymbol{z}_s(t), \boldsymbol{N}_s(t)). \tag{5}$$

with the nonnegativity constraint

$$r_{ij,s}(t) \ge 0. \tag{6}$$

Note that vehicles will be relocated from station i to station j if $F_{ij,t}$ is positive, and that they will be relocated in the reverse direction otherwise. In general, however, this control policy leads to nonconvex optimization.

To overcome this computational difficulty, we consider a linear control policy of the form

$$F_{ij,t}(\boldsymbol{z}_s(t), \boldsymbol{N}_s(t)) = r_{ij}(t) + w_{ij}(t) \cdot \phi_{ij,t}(\boldsymbol{N}_s(t)), \tag{7}$$

with the nonnegativity constraint

$$w_{ij}(t) \geq 0, \tag{8}$$

where $r_{ij}(t)$ and $w_{ij}(t)$ are decision variables, and $\phi_{ij,t}(\boldsymbol{N}_s(t))$ is a feature constructed from $\boldsymbol{N}_s(t)$. Note that $r_{ij}(t)$ corresponds to net relocations regardless of the scenario, and $w_{ij}(t)$ controls relocations based on the history of user trips $(\boldsymbol{N}_s(t))$ in each scenario.

A variety of features can be constructed for $\phi_{ij,t}(\boldsymbol{N}_s(t))$. In particular, we use the following feature in our control policies:

$$\begin{aligned}\phi_{ij,t}(\boldsymbol{N}_s(t)) = & \underbrace{\frac{1}{C_i^{\max}} \sum_{k=t-\tau_{\text{pol}}}^{t-1} \sum_{h \in D} \left(N_{hi,s}^{\text{arr}}(k) - N_{ih,s}^{\text{dep}}(k) \right)}_{\text{increase rate of vehicles at station } i} \\ & - \underbrace{\frac{1}{C_j^{\max}} \sum_{k=t-\tau_{\text{pol}}}^{t-1} \sum_{h \in D} \left(N_{hj,s}^{\text{arr}}(k) - N_{jh,s}^{\text{dep}}(k) \right)}_{\text{increase rate of vehicles at station } j}.\end{aligned} \tag{9}$$

Through Eqs. (5) and (7), this feature increases the number of relocations from station i to station j when the rate of increase in vehicles is greater at station i than at station j during recent τ_{pol} periods.

We also note that vehicle relocations usually have some restrictions mainly because of available staffing levels. Specifically, we suppose that it is possible to launch vehicle relocations in only some periods $t \in T_{\text{rel}} \subseteq T$, implying that

$$r_{ij,s}(t) = 0 \quad (t \in T \setminus T_{\text{rel}}). \tag{10}$$

We also suppose that the number of relocations is limited as

$$\sum_{i \in D} \sum_{j \in D} r_{ij,s}(t) \leq R, \tag{11}$$

where R is the maximum number of vehicles to be relocated in each time period.

2.5. Formulation

There are three objectives to be minimized in our optimization model. The first objective is to reduce the effort required for relocation. Specifically, we minimize the sum of relocated vehicles:

$$\text{RelVeh} := \frac{1}{|S|} \sum_{s \in S} \sum_{t \in T_{\text{rel}}} \sum_{i \in D} \sum_{j \in D} r_{ij,s}(t). \tag{12}$$

The second objective is related to station capacity. Let $a_{j,s}(t)$ calculate violations of station capacity as

$$C_j^{\min} - a_{j,s}(t) \le z_{j,s}(t) \le C_j^{\max} + a_{j,s}(t), \tag{13}$$

$$a_{j,s}(t) \ge 0. \tag{14}$$

We then minimize the sum of such violations:

$$\text{StaCap} := \frac{1}{|S|} \sum_{s \in S} \sum_{t \in T} \sum_{j \in D} a_{j,s}(t). \tag{15}$$

The third objective is to reduce the effort required for returning to the initial number of vehicles at each station at the end of each day. For this purpose, we minimize the sum of deviations from the initial condition as

$$\text{IniCon} := \frac{1}{|S|} \sum_{s \in S} \sum_{j \in D} |z_j^{\text{ini}} - z_{j,s}(t_{\text{end}})|, \tag{16}$$

where t_{end} is the last time period of the day. This objective can be minimized via linear optimization as follows:

$$\text{minimize} \quad \frac{1}{|S|} \sum_{s \in S} \sum_{j \in D} (y_{j,s}^+ + y_{j,s}^-) \tag{17}$$

$$\text{subject to} \quad y_{j,s}^+ - y_{j,s}^- = z_j^{\text{ini}} - z_{j,s}(t_{\text{end}}) \quad (j \in D, s \in S), \tag{18}$$

$$y_{j,s}^+, y_{j,s}^- \ge 0 \quad (j \in D, s \in S), \tag{19}$$

where $y_{j,s}^+$ and $y_{j,s}^-$ are auxiliary decision variables.

We are now in a position to formulate our optimization model to determine linear control policies for online vehicle relocation as follows:

$$\begin{aligned}
\text{minimize} \quad & \frac{\alpha}{|S|}\sum_{s\in S}\sum_{t\in T_{\mathrm{rel}}}\sum_{i\in D}\sum_{j\in D} r_{ij,s}(t) + \frac{\beta}{|S|}\sum_{s\in S}\sum_{t\in T}\sum_{j\in D} a_{j,s}(t) \\
& \quad + \frac{\gamma}{|S|}\sum_{s\in S}\sum_{j\in D}(y^{+}_{j,s} + y^{-}_{j,s}) \qquad (20) \\
\text{subject to} \quad & z_{j,s}(t+1) = z_{j,s}(t) - \sum_{h\in D} N^{\mathrm{dep}}_{jh,s}(t) - \sum_{h\in D} r_{jh,s}(t) + \sum_{h\in D} N^{\mathrm{arr}}_{hj,s}(t) \\
& \quad + \sum_{h\in D} r_{hj,s}(t-\tau_{hj}) \quad (j\in D, s\in S, t\in T\setminus\{t_{\mathrm{end}}\}), \qquad (21) \\
& \sum_{j\in D} z^{\mathrm{ini}}_j = Z, \qquad (22) \\
& z_{j,s}(1) = z^{\mathrm{ini}}_j \quad (j\in D, s\in S), \qquad (23) \\
& r_{ij,s}(t) - r_{ji,s}(t) = r_{ij}(t) + w_{ij}(t)\left(\frac{1}{C^{\max}_i}\sum_{k=t-\tau_{\mathrm{pol}}}^{t-1}\sum_{h\in D}\Big(N^{\mathrm{arr}}_{hi,s}(k) - N^{\mathrm{dep}}_{ih,s}(k)\Big)\right. \\
& \quad \left. -\frac{1}{C^{\max}_j}\sum_{k=t-\tau_{\mathrm{pol}}}^{t-1}\sum_{h\in D}\Big(N^{\mathrm{arr}}_{hj,s}(k) - N^{\mathrm{dep}}_{jh,s}(k)\Big)\right) \quad (i,j\in D, s\in S, t\in T_{\mathrm{rel}}), \\
& \qquad (24) \\
& r_{ij,s}(t) = 0 \quad (i,j\in D, s\in S, t\in T\setminus T_{\mathrm{rel}}), \qquad (25) \\
& \sum_{i\in D}\sum_{j\in D} r_{ij,s}(t) \le R \quad (s\in S, t\in T_{\mathrm{rel}}), \qquad (26) \\
& C^{\min}_j - a_{j,s}(t) \le z_{j,s}(t) \le C^{\max}_j + a_{j,s}(t) \quad (j\in D, s\in S, t\in T), \\
& \qquad (27) \\
& y^{+}_{j,s} - y^{-}_{j,s} = z^{\mathrm{ini}}_j - z_{j,s}(t_{\mathrm{end}}) \quad (j\in D, s\in S), \qquad (28) \\
& a_{j,s}(t) \ge 0 \quad (j\in D, s\in S, t\in T), \qquad (29) \\
& r_{ij,s}(t) \ge 0 \quad (i,j\in D, s\in S, t\in T), \qquad (30) \\
& w_{ij}(t) \ge 0 \quad (i,j\in D, t\in T), \qquad (31) \\
& y^{+}_{j,s}, y^{-}_{j,s} \ge 0 \quad (j\in D, s\in S), \qquad (32) \\
& z^{\mathrm{ini}}_j \ge 0 \quad (j\in D), \qquad (33)
\end{aligned}$$

where α, β, and γ are user-defined weight parameters of the three objectives (12), (15), and (16). This linear optimization problem can be solved to optimality by using optimization software.

3. Computational Experiments

This section reports results from computational experiments designed to evaluate the effectiveness of our method for online vehicle relocation.

3.1. Experimental Design

From Kaggle datasets[3], we downloaded an anonymized dataset of the SF Bay Area Bike Share, which contains over 670,000 bike trips made between August 2013 and August 2015 in the San Francisco Bay Area in the United States. In particular, our experiments cover trips between 35 stations around San Francisco.

Grouping stations into clusters is reasonable for such urban areas with densely distributed stations [7, 8, 12, 13]. As shown in Figure 1[4], we divided the 35 stations into 7 clusters by using the K-means clustering method based on longitude and latitude coordinates of each station.

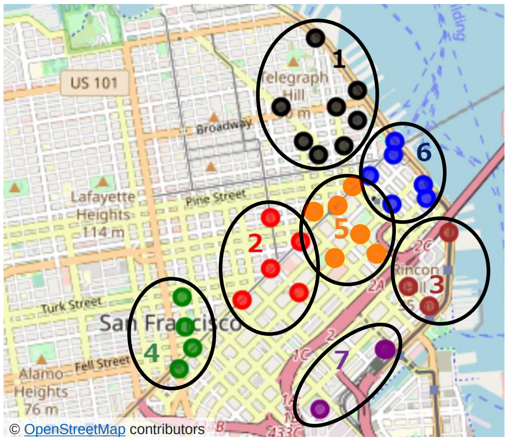


Figure 1: Seven clusters of stations

In our experiments, each station $j \in D$ corresponds to a cluster of stations (i.e., $D = \{1, 2, \ldots, 7\}$). Each period $t \in T$ is 5 min, and the set T of time

[3] `https://www.kaggle.com/benhamner/sf-bay-area-bike-share`

[4] Base map and data from OpenStreetMap and OpenStreetMap Foundation (`www.openstreetmap.org`)

periods corresponds to 24 hours (i.e., $T = \{1, 2, \ldots, 289\}$). The total number of vehicles was set as $Z = 315$. Vehicles are relocated hourly 13 times a day from 08:00 to 20:00 (i.e., $T_{\text{rel}} = \{97, 109, 121, \ldots, 241\}$).

Table 1 lists the station (i.e., station cluster) parameter values, where the column labeled "size" shows the cluster size (i.e., number of stations in a cluster). The maximum number of vehicles ($C_j^{\max}$) was set based on the associated cluster size. The initial number of vehicles (z_j^{ini}) was fixed in our experiments according to the average number of vehicles parked at each station at midnight. The number of time periods required to relocate vehicles was set based on the average trip time as follows:

$$(\tau_{ij})_{(i,j)\in D\times D} = \begin{pmatrix} 0 & 2 & 2 & 4 & 1 & 1 & 3 \\ 2 & 0 & 2 & 2 & 1 & 2 & 2 \\ 2 & 2 & 0 & 4 & 1 & 2 & 1 \\ 3 & 1 & 3 & 0 & 2 & 3 & 2 \\ 2 & 1 & 1 & 2 & 0 & 1 & 2 \\ 1 & 2 & 2 & 3 & 1 & 0 & 2 \\ 3 & 2 & 1 & 3 & 2 & 2 & 0 \end{pmatrix}.$$

Table 1: Parameter values of stations

ID (j)	size	$C_j^{\min}$	$C_j^{\max}$	z_j^{ini}
1	8	0	120	51
2	5	0	95	52
3	3	0	57	33
4	4	0	88	36
5	6	0	122	42
6	6	0	126	69
7	3	0	57	32

Table 2 lists eight problem instances on which training and testing sets were used for optimization and performance evaluation, respectively. Note that $|S|$ is the number of scenarios, and "#trips" is the number of trips. For stress testing, in addition to the original instances (e.g., Aug15-O), we prepared duplicate-trip instances (e.g., Aug15-D) by doubling the number of trips through duplication.

We calculated the following evaluation metrics from the testing sets:

Table 2: Problem instances

	Training set			Testing set		
Instance	Month	$\lvert S\rvert$	#trips	Month	$\lvert S\rvert$	#trips
Nov14-O	October 2014	31	31,019	November 2014	30	23,075
Feb15-O	January 2015	31	25,364	February 2015	28	23,954
May15-O	April 2015	30	28,628	May 2015	31	26,740
Aug15-O	July 2015	31	28,985	August 2015	30	27,479
Nov14-D	October 2014	31	62,038	November 2014	30	46,150
Feb15-D	January 2015	31	50,728	February 2015	28	47,908
May15-D	April 2015	30	57,256	May 2015	31	53,480
Aug15-D	July 2015	31	57,970	August 2015	30	54,958

RelVeh: Sum of relocated vehicles (12),

StaCap: Sum of violations of station capacity (15),

IniCon: Sum of deviations from the initial condition of vehicles (16).

We compare the computational performance of the following methods for dynamic vehicle relocation:

OffRel: Offline relocation method: Solves problem (20)–(33) with the additional constraint

$$w_{ij}(t) = 0 \quad (i, j \in D, t \in T)$$

to execute the same relocation plan for all scenarios.

MPC(E): Model predictive control method based on the expected dynamics [12].

MPC(R): Model predictive control method based on robust interval modelling [13].

LinCon: Our linear control policies: Solves problem (20)–(33), where the number of input time periods was $\tau_{\text{pol}} = 72$ (i.e., six hours).

We implemented these methods in the Python programming language on Google Colaboratory [6] and solved the optimization models using the COIN-OR branch-and-cut solver Cbc[5] [16]. The weight parameters of the objectives were set as $\alpha = \beta = \gamma = 1$. The maximum number of vehicles to be relocated in each time period was set as $R \in \{15, 45\}$ for the original and duplicate-trip problem instances, respectively. The number of relocated vehicles was rounded to the nearest integer, and Eq. (11) was satisfied by means of the largest remainder method [2] if necessary.

3.2. Evaluation of Relocation Performance

Figures 2 and 3 show the evaluation metrics provided by the four relocation methods in the original and duplicate-trip problem instances, respectively.

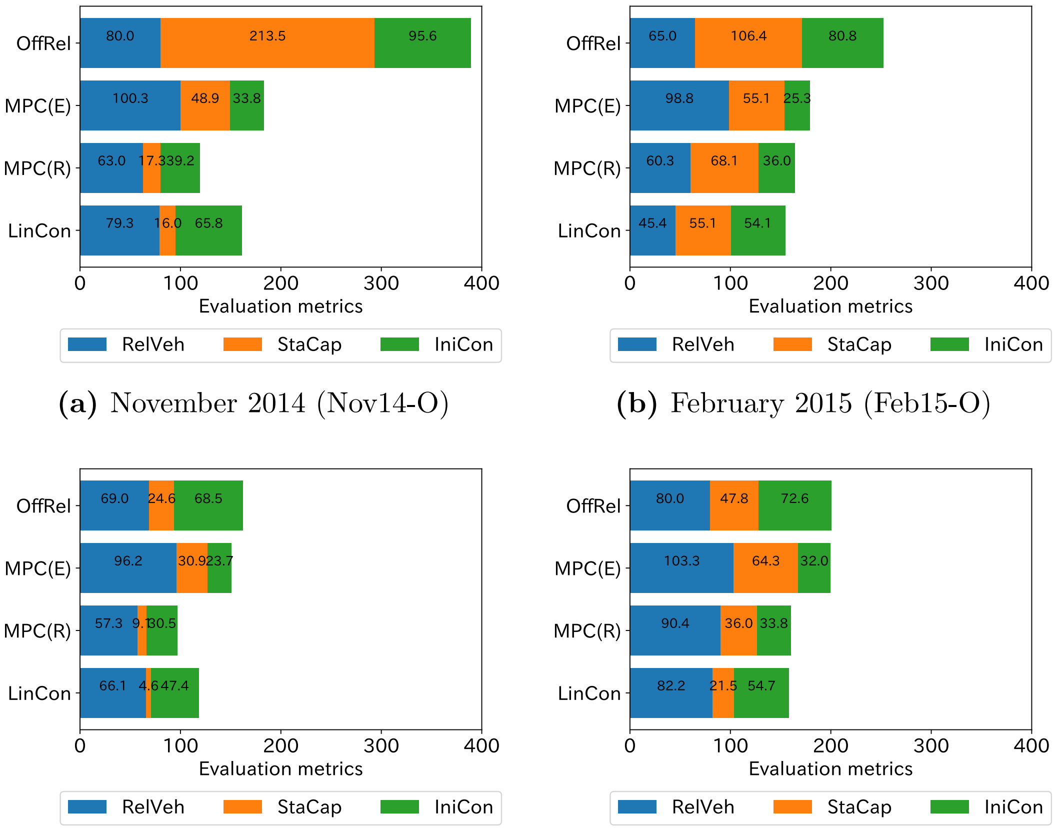
For the original problem instances (Figure 2), our linear control policies (LinCon) and the robust MPC method (MPC(R)) delivered high relocation performance. LinCon performed best in February 2015 (Figure 2b) and August 2015 (Figure 2d), whereas MPC(R) performed best in November 2014 (Figure 2a) and May 2015 (Figure 2c). The performance of the expectation-based MPC method (MPC(E)) was slightly worse than that of LinCon and MPC(R). The performance of the offline relocation method (OffRel) was the worst of the four methods.

The sum of violations of station capacity (StaCap) was often the smallest for LinCon, implying that our linear control policy is highly effective in dynamic vehicle relocation. However, the sum of deviations from the initial condition of vehicles (IniCon) was smaller for the MPC methods (MPC(E) and MPC(R)) than for the other methods. The main reason for this was that the MPC methods repeatedly solve optimization problems based on the latest information for restoring vehicle locations.

For the duplicate-trip problem instances (Figure 3), StaCap was greatly increased because the number of trips was doubled. In this case, the performance of OffRel was by far the worst of the four methods, while the high relocation performance of LinCon remained comparable to that of the MPC methods. These results show that online relocation methods are essential for coping with a large number of trips.

Tables 3 and 4 list the computation times for each relocation method

[5] `https://github.com/coin-or/Cbc`



(a) November 2014 (Nov14-O)

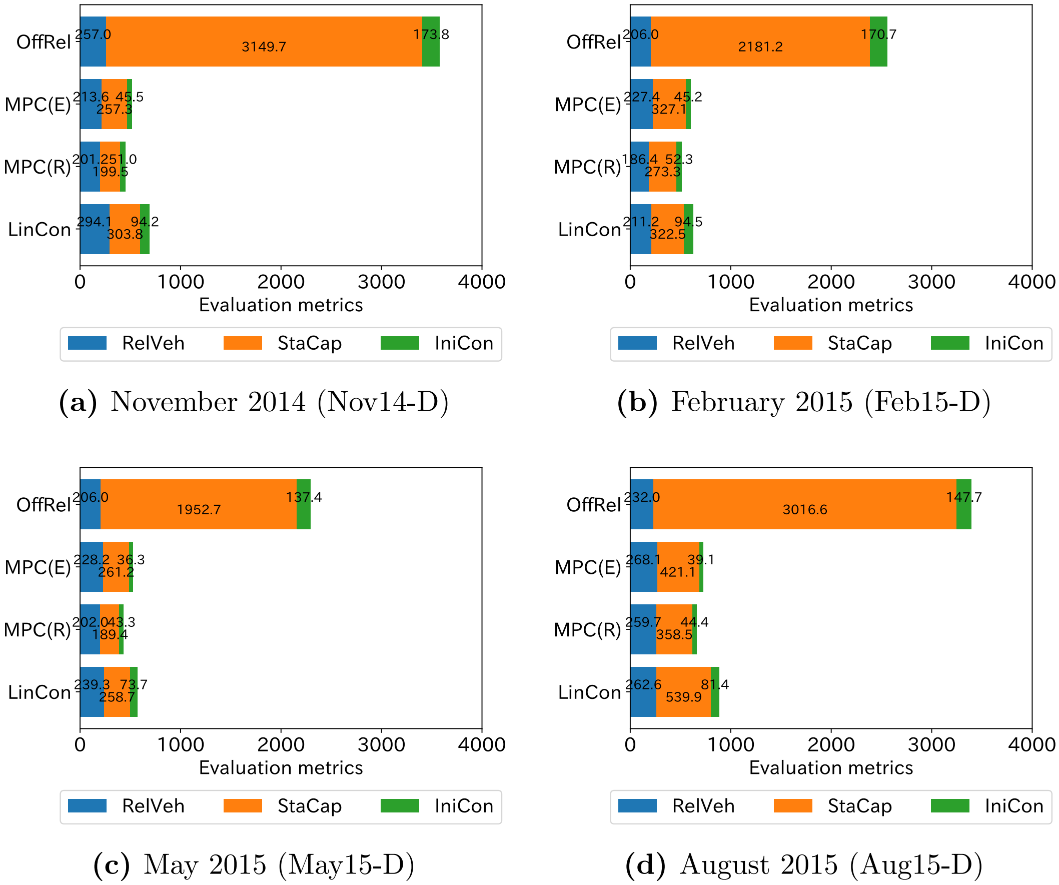
(b) February 2015 (Feb15-O)

(c) May 2015 (May15-O)

(d) August 2015 (Aug15-O)

Figure 2: Relocation performance for the original problem instances

to deal with the original and duplicate-trip problem instances, respectively. "Offline" means the offline computation times required by OffRel and LinCon to solve optimization problems, whereas "Online" means the online computation times required by the four methods to calculate relocations. The MPC methods required a very long computation time to calculate relocations because they solve an optimization problem in every time period. In contrast, OffRel and LinCon were hundreds of times faster than the MPC methods in the online relocation phase. Recall that OffRel completes relocation plans in the offline optimization phase, and that LinCon is capable of calculating relocations quickly from the control policies.



(a) November 2014 (Nov14-D)

(b) February 2015 (Feb15-D)

(c) May 2015 (May15-D)

(d) August 2015 (Aug15-D)

Figure 3: Relocation performance for the duplicate-trip problem instances

3.3. Analysis of Linear Control Policies

We next focus on the results of the Aug15-D instance in order to analyze the validity of our linear control policies. Figures 4 and 5 show box plots of the number of vehicles at each station in August 2015 for the Aug15-D instance. The figures on the left show the results without relocations, and the figures on the right show the results obtained by linear control policies. Note that two parallel solid lines in each figure represent $C_j^{\min}$ and $C_j^{\max}$, the minimum and maximum numbers of vehicles at station $j \in D$ (see also Table 1).

The number of vehicles often exceeded the station capacity when no relocations were performed (Figures 4a, 5a, and 5e). In contrast, our linear

Table 3: Computation times [s] for the original problem instances

		Problem instance			
	Method	Nov14-O	Feb15-O	May15-0	Aug15-O
Offline	OffRel	13.0	11.1	10.8	11.6
	LinCon	57.4	43.0	44.4	51.3
Online	OffRel	2.0	1.8	2.0	1.9
	MPC(E)	483.3	364.7	402.4	401.7
	MPC(R)	1866.1	1708.2	1641.8	1600.0
	LinCon	3.8	3.9	4.0	4.0

Table 4: Computation times [s] for the duplicate-trip problem instances

		Problem instance			
	Method	Nov14-D	Feb15-D	May15-D	Aug15-D
Offline	OffRel	23.0	20.2	17.5	17.4
	LinCon	154.4	79.9	106.4	134.1
Online	OffRel	1.6	1.5	1.7	2.5
	MPC(E)	367.5	355.8	386.9	376.6
	MPC(R)	1589.2	1457.8	1675.5	1602.5
	LinCon	3.4	3.1	3.5	3.4

control policies greatly reduced violations of station capacity, so the number of vehicles often fell between the minimum to the maximum at stations (Figures 4b, 5b, and 5f).

Figure 6 show heat maps of net relocations $(r_{ij}(t))_{(i,j)\in D\times D}$ and input weights $(w_{ij}(t))_{(i,j)\in D\times D}$ of control policies at 08:00, 12:00, and 16:00 in August 2015 for the Aug15-D instance.

At 08:00, vehicles are mainly relocated from stations 1 and 5 to station 7 (Figure 6a). Indeed, if such relocations are not performed, stations 1 and 5 become full and station 7 becomes empty (Figures 4a, 5a, and 5e).

At 12:00, the relocation plans were not pre-determined (Figure 6c) but varied depending on the situation. For example, Figure 6d shows that if the number of vehicles at station 1 increases, vehicles should be relocated to stations 3, 4, and 5. We can also see that if the number of vehicles at station 2 decreases, vehicles should be relocated from stations 3, 5, and 6.

At 16:00, vehicles are mainly relocated from station 7 to station 1 (Figure 6e). Note that station 7 becomes full without such relocations (Figure 5e). In addition, vehicles are likely to be relocated from station 7 to stations 3 and 4 depending on the situation (Figure 6f).

4. Conclusion

This paper addressed the vehicle relocation problem in one-way station-based shared mobility systems. For this problem, we devised linear control policies based on a time-space network to determine online vehicle relocations from a history of user trips. Our scenario-based optimization model was formulated as a linear optimization problem.

To confirm the effectiveness of our method, we conducted computational experiments using a real-world dataset from the SF Bay Area Bike Share. Our linear control policies were comparable in relocation performance to MPC-based strategies [12, 13]. In addition, our method was much faster than MPC methods in the online relocation phase. We also examined how our relocation strategies improved the occupancy state of each station.

Although the advantages of optimization and simulation models are combined in multistage approaches [24, 26, 33, 36], solving optimization models in the simulation process is computationally expensive. We successfully overcame this challenge by introducing linear control methods to the vehicle relocation problem while maintaining high relocation performance. Once the optimization model has been solved, the obtained control policies can easily be put into practical use without complicated online calculations in shared mobility systems.

A future direction of study will be to use nonlinear control policies [41, 42] with the aim of improving the relocation performance. One promising approach is to introduce a risk measure into the vehicle relocation problem [46]. Another direction for future research is to more accurately reflect reality in the optimization model. For instance, the relocation staff uses trucks in many bikesharing systems [19, 29, 30, 32].

Acknowledgments

This work was partially supported by JSPS KAKENHI Grant Number JP21K04526.

References

[1] Ataç, S., Obrenović, N., & Bierlaire, M. (2021). Vehicle Sharing Systems: A Review and a Holistic Management Framework. EURO Journal on Transportation and Logistics, 100033.

[2] Balinski, M. L., & Young, H. P. (2010). Fair representation: Meeting the ideal of one man, one vote. Brookings Institution Press.

[3] Barth, M., & Todd, M. (1999). Simulation model performance analysis of a multiple station shared vehicle system. Transportation Research Part C: Emerging Technologies, 7(4), 237–259.

[4] Ben-Tal, A., Goryashko, A., Guslitzer, E., & Nemirovski, A. (2004). Adjustable robust solutions of uncertain linear programs. Mathematical programming, 99(2), 351–376.

[5] Bertsimas, D., & Goyal, V. (2012). On the power and limitations of affine policies in two-stage adaptive optimization. Mathematical programming, 134(2), 491–531.

[6] Bisong, E. (2019). Google colaboratory. In Building Machine Learning and Deep Learning Models on Google Cloud Platform (pp. 59–64). Apress, Berkeley, CA.

[7] Boyacı, B., Zografos, K. G., & Geroliminis, N. (2015). An optimization framework for the development of efficient one-way car-sharing systems. European Journal of Operational Research, 240(3), 718–733.

[8] Boyacı, B., Zografos, K. G., & Geroliminis, N. (2017). An integrated optimization-simulation framework for vehicle and personnel relocations of electric carsharing systems with reservations. Transportation Research Part B: Methodological, 95, 214–237.

[9] Brendel, A. B., Lichtenberg, S., Brauer, B., Nastjuk, I., & Kolbe, L. M. (2018). Improving electric vehicle utilization in carsharing: A framework and simulation of an e-carsharing vehicle utilization management system. Transportation Research Part D: Transport and Environment, 64, 230–245.

[10] Calafiore, G. C. (2008). Multi-period portfolio optimization with linear control policies. Automatica, 44(10), 2463–2473.

[11] Calafiore, G. C. (2015). Optimal dynamic asset allocation with lower partial moments criteria and affine policies. International Journal of Financial Engineering and Risk Management, 2(2), 87–108.

[12] Calafiore, G. C., Bongiorno, C., & Rizzo, A. (2018, December). A control-oriented model for mobility on demand systems. In 2018 IEEE Conference on Decision and Control (CDC) (pp. 999-1004). IEEE.

[13] Calafiore, G. C., Bongiorno, C., & Rizzo, A. (2019). A robust MPC approach for the rebalancing of mobility on demand systems. Control Engineering Practice, 90, 169–181.

[14] Camacho, E. F., & Bordons, C. (2007). Model Predictive Control. Springer-Verlag.

[15] Ferrero, F., Perboli, G., Rosano, M., & Vesco, A. (2018). Car-sharing services: An annotated review. Sustainable Cities and Society, 37, 501–518.

[16] Forrest, J., & Lougee-Heimer, R. (2005). CBC user guide. In Emerging theory, methods, and applications (pp. 257–277). INFORMS.

[17] Garstka, S. J., & Wets, R. J. B. (1974). On decision rules in stochastic programming. Mathematical Programming, 7(1), 117–143.

[18] Gavalas, D., Konstantopoulos, C., & Pantziou, G. (2016). Design and management of vehicle-sharing systems: A survey of algorithmic approaches. Smart Cities and Homes, 261–289.

[19] Ghosh, S., Varakantham, P., Adulyasak, Y., & Jaillet, P. (2017). Dynamic repositioning to reduce lost demand in bike sharing systems. Journal of Artificial Intelligence Research, 58, 387–430.

[20] Illgen, S., & Höck, M. (2018). Electric vehicles in car sharing networks—Challenges and simulation model analysis. Transportation Research Part D: Transport and Environment, 63, 377-387.

[21] Illgen, S., & Höck, M. (2019). Literature review of the vehicle relocation problem in one-way car sharing networks. Transportation Research Part B: Methodological, 120, 193–204.

[22] Jorge, D., Correia, G. H., & Barnhart, C. (2014). Comparing optimal relocation operations with simulated relocation policies in one-way carsharing systems. IEEE Transactions on Intelligent Transportation Systems, 15(4), 1667–1675.

[23] Jorge, D., Molnar, G., & de Almeida Correia, G. H. (2015). Trip pricing of one-way station-based carsharing networks with zone and time of day price variations. Transportation Research Part B: Methodological, 81, 461–482.

[24] Kaspi, M., Raviv, T., Tzur, M., & Galili, H. (2016). Regulating vehicle sharing systems through parking reservation policies: Analysis and performance bounds. European Journal of Operational Research, 251(3), 969–987.

[25] Katzev, R. (2003). Car sharing: A new approach to urban transportation problems. Analyses of Social Issues and Public Policy, 3(1), 65–86.

[26] Kek, A. G., Cheu, R. L., Meng, Q., & 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), 149–158.

[27] Kek, A. G., Cheu, R. L., & Chor, M. L. (2006). Relocation simulation model for multiple-station shared-use vehicle systems. Transportation research record, 1986(1), 81–88.

[28] Laporte, G., Meunier, F., & Calvo, R. W. (2018). Shared mobility systems: An updated survey. Annals of Operations Research, 271(1), 105–126.

[29] Li, Y., Zheng, Y., & Yang, Q. (2018, July). Dynamic bike reposition: A spatio-temporal reinforcement learning approach. In Proceedings of the

24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (pp. 1724–1733).

[30] 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), 740–753.

[31] Nair, R., & Miller-Hooks, E. (2011). Fleet management for vehicle sharing operations. Transportation Science, 45(4), 524–540.

[32] Nath, R. B., & Rambha, T. (2019). Modelling methods for planning and operation of bike-sharing systems. Journal of the Indian Institute of Science, 99(4), 621–645.

[33] Nourinejad, M., & Roorda, M. J. (2014). A dynamic carsharing decision support system. Transportation research part E: logistics and transportation review, 66, 36–50.

[34] Nourinejad, M., Zhu, S., Bahrami, S., & Roorda, M. J. (2015). Vehicle relocation and staff rebalancing in one-way carsharing systems. Transportation Research Part E: Logistics and Transportation Review, 81, 98–113.

[35] Pfrommer, J., Warrington, J., Schildbach, G., & Morari, M. (2014). Dynamic vehicle redistribution and online price incentives in shared mobility systems. IEEE Transactions on Intelligent Transportation Systems, 15(4), 1567–1578.

[36] Repoux, M., Boyacı, B., & Geroliminis, N. (2015, January). Simulation and optimization of one-way car-sharing systems with variant relocation policies. In 94th Annual Meeting of the Transportation Research Board.

[37] Repoux, M., Kaspi, M., Boyacı, B., & 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, 82–104.

[38] Shaheen, S., Cohen, A., Chan, N., & Bansal, A. (2020). Sharing strategies: carsharing, shared micromobility (bikesharing and scooter sharing), transportation network companies, microtransit, and other innovative mobility modes. In Transportation, Land Use, and Environmental Planning (pp. 237–262). Elsevier.

[39] Shapiro, A., & Nemirovski, A. (2005). On complexity of stochastic programming problems. In Continuous optimization (pp. 111–146). Springer, Boston, MA.

[40] Skaf, J., & Boyd, S. P. (2010). Design of affine controllers via convex optimization. IEEE Transactions on Automatic Control, 55(11), 2476–2487.

[41] Takano, Y., & Gotoh, J. (2011). A nonlinear control policy using kernel method for dynamic asset allocation. Journal of the Operations Research Society of Japan, 54(4), 201–218.

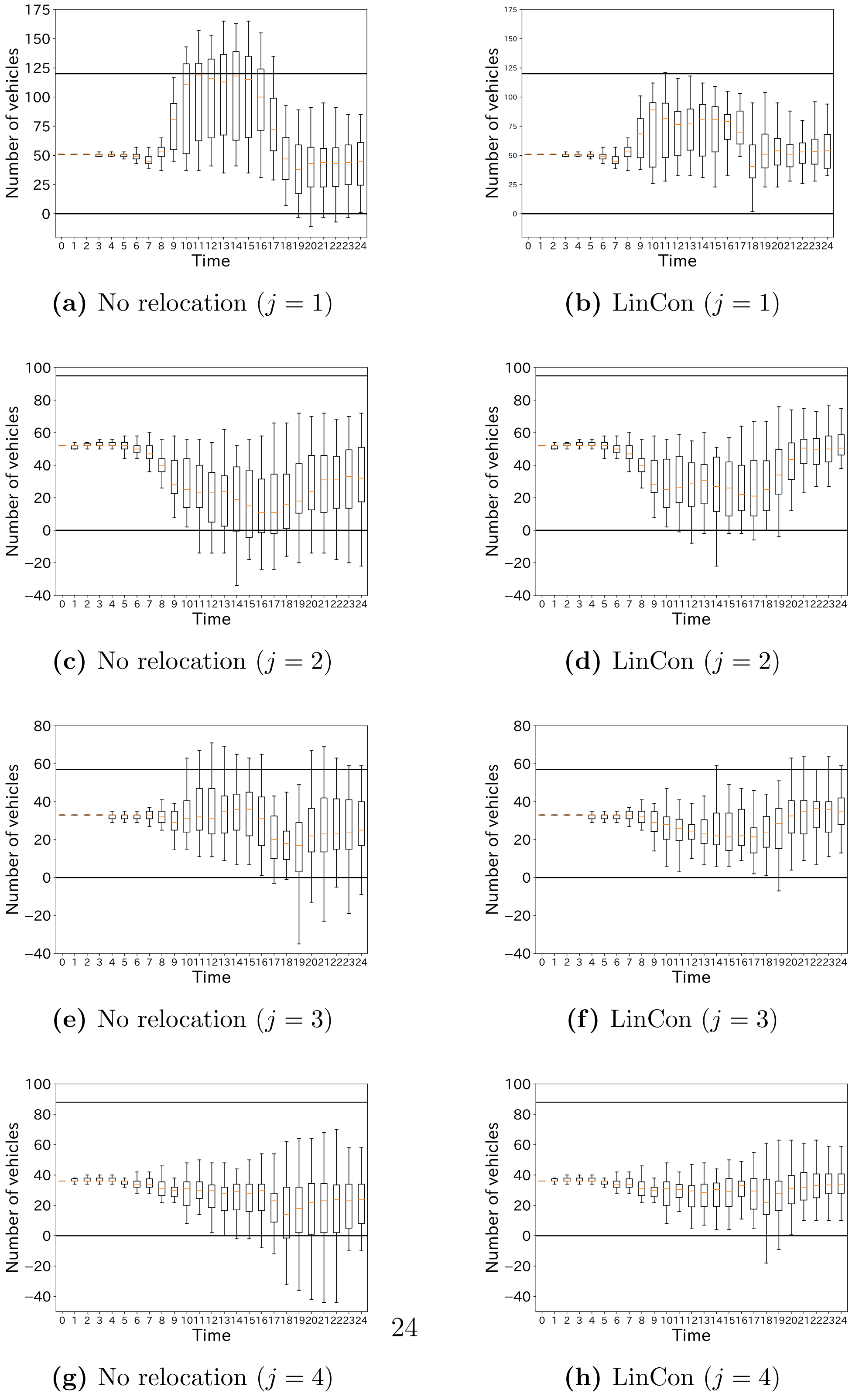
[42] Takano, Y., & Gotoh, J. (2014). Multi-period portfolio selection using kernel-based control policy with dimensionality reduction. Expert systems with applications, 41(8), 3901–3914.

[43] Takano, Y., & Gotoh, J. (2019). Dynamic portfolio selection with linear control policies for coherent risk minimization. Optimization Online.

[44] Warrington, J., & Ruchti, D. (2019). Two-stage stochastic approximation for dynamic rebalancing of shared mobility systems. Transportation Research Part C: Emerging Technologies, 104, 110–134.

[45] Waserhole, A., & Jost, V. (2016). Pricing in vehicle sharing systems: Optimization in queuing networks with product forms. EURO Journal on Transportation and Logistics, 5(3), 293–320.

[46] Zhang, K., Takano, Y., Wang, Y., & Yoshise, A. (2021). Optimizing the strategic decisions for one-way station-based carsharing systems: A mean-CVaR approach. IEEE Access.



(a) No relocation ($j = 1$)

(b) LinCon ($j = 1$)

(c) No relocation ($j = 2$)

(d) LinCon ($j = 2$)

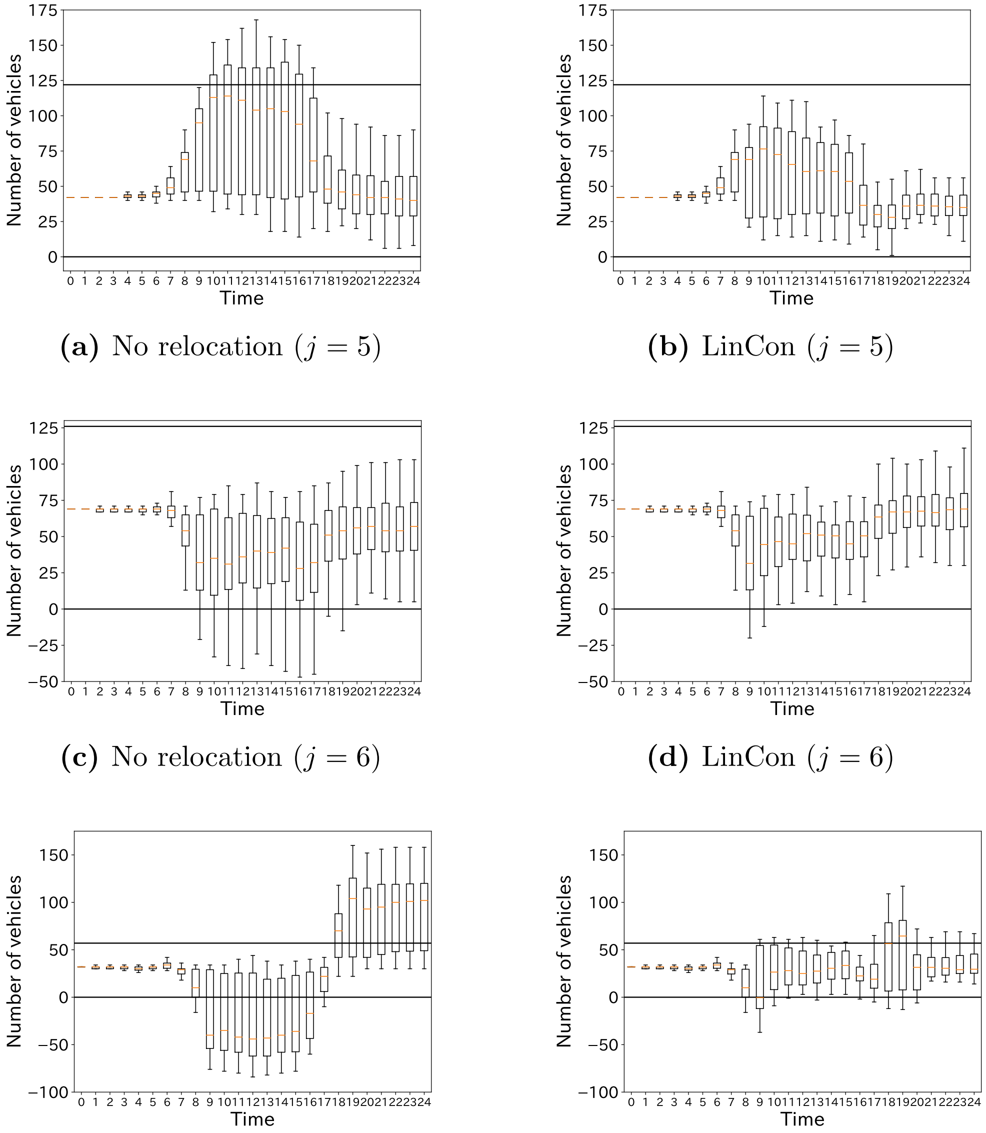
(e) No relocation ($j = 3$)

(f) LinCon ($j = 3$)

(g) No relocation ($j = 4$)

(h) LinCon ($j = 4$)

Figure 4: Changes in the number of vehicles at each station $j \in \{1, 2, 3, 4\}$ in August 2015 (Aug15-D)



(a) No relocation ($j = 5$)

(b) LinCon ($j = 5$)

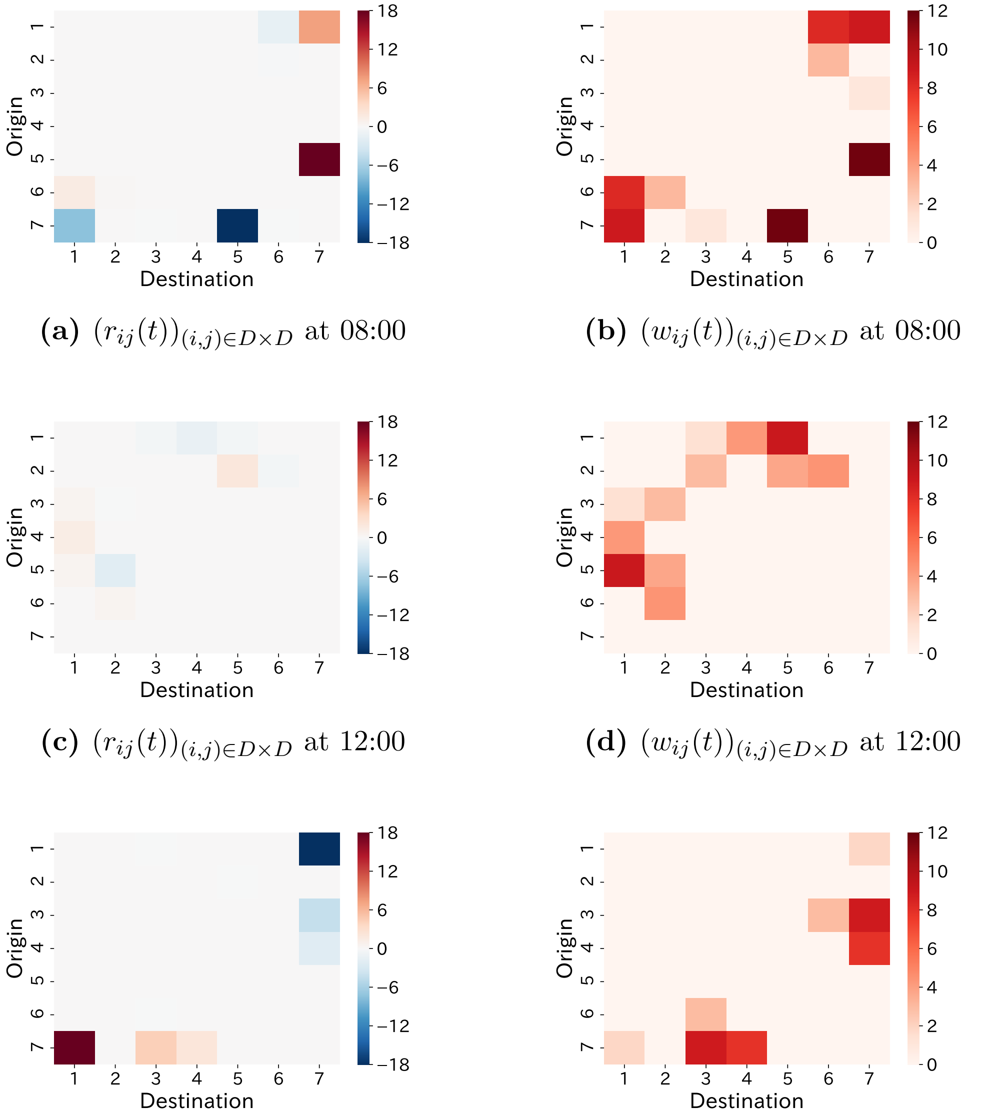
(c) No relocation ($j = 6$)

(d) LinCon ($j = 6$)

(e) No relocation ($j = 7$)

(f) LinCon ($j = 7$)

Figure 5: Changes in the number of vehicles at each station $j \in \{5, 6, 7\}$ in August 2015 (Aug15-D)



(a) $(r_{ij}(t))_{(i,j)\in D\times D}$ at 08:00

(b) $(w_{ij}(t))_{(i,j)\in D\times D}$ at 08:00

(c) $(r_{ij}(t))_{(i,j)\in D\times D}$ at 12:00

(d) $(w_{ij}(t))_{(i,j)\in D\times D}$ at 12:00

(e) $(r_{ij}(t))_{(i,j)\in D\times D}$ at 16:00

(f) $(w_{ij}(t))_{(i,j)\in D\times D}$ at 16:00

Figure 6: Heat maps illustrating linear control policies in August 2015 (Aug15-D)