Les Cahiers du GERAD ISSN: 0711–2440

Dynamic rebalancing optimization for bike-sharing systems: A modeling framework and empirical comparison

J. Liang, S. D. Jena, A. Lodi

G–2023–47

October 2023

Citation suggérée : J. Liang, S. D. Jena, A. Lodi (Octobre 2023). Dynamic rebalancing optimization for bike-sharing systems: A modeling framework and empirical comparison, Rapport technique, Les Cahiers du GERAD G– 2023–47, GERAD, HEC Montréal, Canada.

Avant de citer ce rapport technique, veuillez visiter notre site Web (https://www.gerad.ca/fr/papers/G-2023-47) afin de mettre à jour vos données de référence, s'il a été publié dans une revue scientifique.

Suggested citation: J. Liang, S. D. Jena, A. Lodi (October 2023). Dynamic rebalancing optimization for bike-sharing systems: A modeling framework and empirical comparison, Technical report, Les Cahiers du GERAD G–2023–47, GERAD, HEC Montréal, Canada.

Before citing this technical report, please visit our website (https://www.gerad.ca/en/papers/G-2023-47) to update your reference data, if it has been published in a scientific journal.

La publication de ces rapports de recherche est rendue possible grâce au soutien de HEC Montréal, Polytechnique Montréal, Université McGill, Université du Québec à Montréal, ainsi que du Fonds de recherche du Québec – Nature et technologies.

Dépôt légal – Bibliothèque et Archives nationales du Québec, 2023
– Bibliothèque et Archives Canada, 2023

The publication of these research reports is made possible thanks to the support of HEC Montréal, Polytechnique Montréal, McGill University, Université du Québec à Montréal, as well as the Fonds de recherche du Québec – Nature et technologies.

Legal deposit – Bibliothèque et Archives nationales du Québec, 2023
– Library and Archives Canada, 2023

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

Dynamic rebalancing optimization for bike-sharing systems: A modeling framework and empirical comparison

Jiaqi Liang [a, b, e]

Sanjay Dominik Jena [c]

Andrea Lodi [b, d, e]

[a] *Polytechnique Montréal, Montréal (Qc), Canada, H3T 1J4*

[b] *GERAD, Montréal (Qc), Canada, H3T 1J4*

[c] *Université du Quèbec à Montréal, Montréal (Qc), Canada, H2X 3X2*

[d] *Cornell Tech and Technion - IIT, New York, USA, 10044*

[e] *Canada Excellence Research Chair in Data-Science for Real-time Decision-Making, Montréal (Qc), Canada, H3T 1J4*

jiaqi.liang@polymtl.ca
jena.sanjay-dominik@uqam.ca
andrea.lodi@cornell.edu

October 2023
Les Cahiers du GERAD
G–2023–47

Abstract : Station-based Bike-sharing systems have been implemented in multiple major cities, offering a low-cost and environmentally friendly transportation alternative. As a remedy to unbalanced stations, operators typically rebalance bikes by trucks. The resulting dynamic planning has received significant attention from the Operations Research community. Due to its modeling flexibility, mixed-integer programming remains a popular choice. However, the complex planning problem requires significant simplifications to obtain a computationally tractable model. As a result, existing models have used a large variety of modeling assumptions and techniques regarding decision variables and constraints. Unfortunately, the impact of such assumptions on the solutions' performance in practice remains generally unexplored.

In this paper, we first systematically survey the literature on rebalancing problems and their modeling assumptions. We then propose a general mixed-integer programming model for multi-period rebalancing problems that can be easily adapted to different assumptions, including trip modeling, time discretization, trip distribution, and event sequences. We develop an instance generator to synthesize realistic station networks and customer trips, as well as a realistic fine-grained simulator to evaluate the operational performance of rebalancing strategies. Finally, extensive numerical experiments are carried out, both on the synthetic and on real-world data, to analyze the effectiveness of various modeling assumptions and techniques. Based on our results, we identify the assumptions that empirically provide the most effective rebalancing strategies in practice. Specifically, a set of specific trip distribution constraints and event sequences ignored in the previous literature seem to provide particularly good results.

Keywords: Facilities planning and design, bike sharing systems, dynamic rebalancing, modeling framework, mixed-integer programming

Acknowledgements: The authors are thankful to BIXI Montreal for providing real-world trip data and for their support throughout several discussions. We also thank Compute Canada for providing the computational resources to carry out the numerical experiments. The work of the second author was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada under grant 2017-05224. We are indebted to three anonymous referees whose detailed comments helped us to significantly improve the paper.

1 Introduction

Bike-sharing systems (BSSs) are quickly gaining popularity worldwide, as they help to reduce traffic congestion and vehicle CO_2 emissions. Over the past few years, BSSs were implemented in most major cities such as New York, Boston, London, Sydney, Beijing, Paris, Toronto, and Montreal. We focus on station-based BSSs, where stations with specific capacities are installed in the city, holding an inventory of rentable bikes. Users may rent available bikes from these stations and return their bikes to available docks.

Throughout the day, BSS stations are often unbalanced, which leads to demand unsatisfaction, given that rental demand may not be satisfied when a station is empty, and return demand may not be satisfied when the station is full (i.e., no empty docks are available). As a remedy, BSS operators employ trucks to rebalance bikes among stations. However, due to the uncertain rental and return demand, as well as the complexity of the dynamic planning problem, manual planning tends to be sub-optimal.

To provide decision support, the scientific community has provided a large variety of predictive and prescriptive models, aiming at improving station rebalancing. Planning models can roughly be divided into those based on Markov Decision Process (MDP) and those based on Mixed-integer Linear Programming (MILP). The former, more recently applied in the context of BSSs [see, e.g., 1, 2, 3, 4, 5, 6, 7], implicitly considers uncertainty. However, in order to remain computationally tractable, MDP models rely on state and action spaces limited in size and are therefore constrained to small station networks, small rebalancing fleets, or limited rebalancing decisions. The majority of the literature proposes MILP models to represent the planning problem, where MILP remains the predominant modeling tool due to its flexibility, as well as the availability of a well-established process to integrate and maintain such models and a wide range of solution methods within an industrial decision-making process. For MILP, both deterministic customer demands [see, e.g., 8, 9, 10] and simplified variants dealing with uncertain demand [see, e.g., 11, 12, 13, 14] have been considered in the literature. In both cases, due to the complexity of the planning problem and the synergies between customer arrivals and rebalancing operations, its modeling requires assumptions that greatly vary within the literature. These assumptions range from the planning objective to the actual decisions and practical constraints used within the models. Further, customer demands may occur continuously in time. To remain computationally feasible, the planning horizon is typically divided into a set of time-periods, which raises questions concerning the sequence of occurring rental and return demands, as well as the moment of the planned rebalancing operations. As a result, both the planning problems and the mathematical optimization models proposed in the literature are highly diverse. Unfortunately, most of those works have been proposed isolated from the remainder of the literature, which therefore lacks consensus on which assumptions and modeling techniques are best to use. Thus, operators are rather uncertain about which modeling assumptions should be used in the context of their specific BSS and which modeling techniques provide solutions that perform best in practice.

Objective, scope, and contributions. The objective of this paper is to provide guidelines to both practitioners and academics on which assumptions and techniques are most appropriate and likely to produce rebalancing solutions that perform well in practice. To this end, we provide a systematic review of the modeling assumptions and techniques presented in the literature, mostly focused on multi-period models, and propose a general MILP modeling framework that encapsulates most of the relevant modeling assumptions. While several modeling assumptions have been used in the literature without further justification or comparison, we explicitly discuss the alternatives and provide intuitive insights on which assumptions may be more appropriate in practice. We then provide extensive numerical results to empirically evaluate the realism of the various assumptions and modeling techniques, such as variable domains, time discretization, the distribution of trip demand, and the assumed sequence of bike rentals, bike returns, and rebalancing operations.

We develop a realistic simulator that emulates customer trips and the given rebalancing strategy on a minute-to-minute basis. This simulator evaluates the quality of a proposed rebalancing planning solution and hence the realism of the modeling assumptions and techniques of the associated optimization model. Throughout our experiments, the most relevant combinations of modeling assumptions and techniques are then empirically tested on a large set of synthetically generated problem instances that have been carefully designed to represent different BSS settings and fit the demand patterns observed in real-world trip data from BIXI Montreal.

Based on our modeling framework and empirical results on both synthetic and real-world data, practitioners can derive an optimization model tailored to their BSS environment. In particular, our empirical results suggest that (i) both variable domain and type strongly impact the realism and tractability of the model, (ii) time-related constraints are particularly important when time-periods are short, (iii) a new set of trip distribution constraints performs better than those previously considered in the literature, and (iv) the sequences of trips and rebalancing operations used in the literature are outperformed by the new event sequences proposed in our paper.

We note again that we are less concerned with the possible uncertainty surrounding the input parameters of the models, but rather with the underlying assumptions required to tractably represent the reality within the mathematical model. As such, the presented modeling framework is deterministic (i.e., it uses a single set of expected trip demand) for several reasons. Stochastic, scenario-based formulations are equally subject to such modeling assumptions. Our conclusions are therefore still likely to hold for such problem variants. While we review and discuss the corresponding literature, for the purpose of our study, it is not necessary to explicitly represent uncertainty. Finally, a stochastic problem variant would be computationally intractable, thus forbidding us to obtain close-to-optimal solutions that are required for our analysis.

Outline. The rest of this paper is organized as follows. Section 2 reviews related literature on BSS rebalancing problems and summarizes the assumptions. Section 3 describes the modeling framework for the multi-period rebalancing problem, including a basic model that can be extended by several constraints according to the various modeling assumptions. Section 4 presents the general framework used to evaluate the practical performance of a given multi-period rebalancing strategy. Numerical tests and analyses on synthetic and real-world problem instances are illustrated in Section 5. This is followed by the conclusions in Section 6.

2 A Review of BSS rebalancing modeling assumptions

The literature mainly focuses on two types of rebalancing in BSSs [15]. User-based rebalancing incentivizes users to rent or return bikes at specific stations [16]. Such an approach is more common in dockless BSSs. In contrast, operator-based rebalancing involves the active management of a rebalancing fleet (e.g., trucks) that relocates bikes from one station to another. Such an approach is specific to station-based BSSs [17]. According to a recent statistical report [18], station-based systems are, by far, more common. Even in the case of rebalancing planning in dockless BSSs, it has been a common practice to divide the studying area into different sub-areas, which are then seen as stations [see, e.g., 5, 19, 20, 21]. We, therefore, focus on operator-based rebalancing.

Operator-based bike-sharing rebalancing problems can be divided into static bicycle repositioning problems (SBRP) and dynamic bicycle repositioning problems (DBRP) [22, 23]. SBRP typically rebalance stations overnight, while the operations during the day are not considered. Static rebalancing, therefore, prepares the station inventories for the next day. However, it cannot explicitly react to the demand fluctuation occurring during the day. In contrast, DBRP focus on intraday rebalancing, where customer trips carried out during the day heavily affect the availability of bikes and docks [24]. Indeed, demand satisfaction highly depends on the real-time status of the stations and customers' stochastic rentals and returns, which poses practical challenges [25]. We here focus on dynamic rebalancing planning, which has a higher impact on practice since it considers continuous rebalancing throughout the day. Given that DBRP consider trip demand and rebalancing operations over the day, the models proposed in the literature either approach this problem using a repeatedly solved single-period model or a multi-period model.

Single-period rolling vs. multi-period planning. A single-period model generally spans a duration between 10 and 60 minutes. Single-period models are typically solved in a rolling horizon fashion throughout the day, while multi-period models are either solved once at the beginning of the planning horizon or several times throughout the day.

Solving a single-period model is computationally easier than a multi-period one. However, single-period models tend to be myopic, i.e., the decisions made at the current time-period cannot take into consideration the consequences in future time-periods. Such models therefore greedily maximize demand satisfaction at the current time-period, even if this results in station inventories that cannot satisfy demand well in the next time-periods. In contrast, multi-period models benefit from integrated planning over all time-periods and avoid suffering from myopic behavior. On the downside, these models may be more difficult to solve, given that they consider several sets of decision variables and constraints for each time-period.

Since we are concerned with finding the model that performs best in practice, we focus on the multi-period planning model. In the following, we summarize the main assumptions made in multi-period planning models proposed in the literature, review existing alternatives and propose some that might not have been considered yet.

2.1 Time discretization and time constraint

To represent the change in stations' status and vehicles, the planning horizon is discretized into time-periods. One mainly has two possibilities to discretize the planning horizon.

- **Time-periods of equal length**. The planning horizon is discretized into a set of evenly-spaced time points and the length of each period is the same, which is employed in most multi-period models. In the multi-period planning framework, we could obtain the changes of stations for each time-period and gain the look-ahead information. However, it is hard to define the optimal length of the time-period. It should depend on the model complexity and the length of the planning horizon we focus on.
- **Time-periods of different lengths**. Here, the length of each period in the planning horizon can be different. [10] split each cumulated demand function into weakly-monotonic segments with extreme values that are regarded as end-of-segment events. Further, the arrival of a vehicle at a station to rebalance bikes is referred to as a station-visit event. These two types of events are sequentially considered to separate the planning horizon.

For multi-period dynamic rebalancing, time-period with equal length is most common in the literature [see, e.g., 14, 26, 27]. Typically, it is assumed that each vehicle rebalances at most one station during one time-period. Selecting an appropriate length of time-period is crucial to the rebalancing strategy and its performance. Several works have investigated the effect of such aggregation on the rebalancing performance [see, e.g. 9, 14, 28]. Generally, aggregations over shorter time-periods allow for more rebalancing operations and, therefore, lower lost demand. However, this typically comes at the cost of increased computing times.

Moreover, when the time-period is short, a time constraint may be required to ensure that the required time for rebalancing and transiting to the next station fits into the length of the time-period. Time constraints, therefore, interdict truck relocations that are unrealistic in practice. [9] and [29] do not use time constraints, while follow-up work [13, 30] apply time constraints within a single-period rolling planning framework. [8] and [14] use time constraints considering only vehicle traveling time, while the handling time at stations is ignored. [10] and [31] consider both traveling and handling time, where the latter is computed as an average value regardless of the number of loading/unloading operations. A summary of how time constraints are handled in the literature, along with other model characteristics, is presented in Table 14 in the Online Supplement.

Note that the introduction of time constraints may make the model difficult to solve. It is therefore crucial to select a proper time-period length that can lead to a reasonable solution time while providing high-quality rebalancing strategies.

2.2 Trip modeling and variable domains

Each customer trip contains one bike rental demand and one bike return demand. To model successful trips, mainly two types of variables have been considered:

- **Origin-destination (O-D) variables** $x_{s_1,s_2}^{t_1,t_2}$, contain departure station s_1, arrival station s_2, departure time t_1, and arrival time t_2.
- **Station-based variables** represent the departure of a trip, i.e., satisfied rental demand $x_{s_1}^{+,t_1}$ and the arrival of a trip, i.e., satisfied return demand $x_{s_2}^{-,t_2}$.

Station-based variable models require fewer variables but lack the connection between rentals and returns. The O-D variable models avoid this issue at the expense of a large number of variables, which may complicate the solutions of the model. A general classification of existing models can be found in Table 15 in the Online Supplement.

Next to the type of variables, the variable domains may also impact the performance of the induced solutions. In the rebalancing model, variables represent three main actions: routing, rebalancing, and the above-discussed user trips. The routing variables typically indicate the location of a vehicle along the planning horizon and the route taken. These variables are always binary. Rebalancing variables represent the inventory of vehicles and the number of bikes to be rebalanced. Most models define them as continuous variables [see, e.g., 8, 10, 13, 14, 29, etc.]. [31] use integer rebalancing variables. User trip variables for rentals and returns also interact with the inventories of stations. Most models in the literature use continuous variables, except for [32], which use integer O-D variables.

2.3 Trip distribution

If the rental/return demand exceeds the current inventory of bikes/docks, a basic optimization model (such as the one in Section 3.1) may select the rentals/returns opportunistically according to the objective function. In reality, however, demand will be satisfied based on a first-come-first-serve rule. Several works therefore aimed at enforcing a more realistic distribution of the trips by adding specific constraints. We review the existing assumptions on demand distribution below:

- **Station-based variables without distribution**. Two variables are created to present rentals and returns for each station and each period [see, e.g., 10]. Similarly, [8] use two variables to represent the shortage and excess of bikes, which is equivalent to the use of station-based variables. Demands will be greedily satisfied in the optimization model without considering the link between rentals and returns. As a result, the lost demand tends to be underestimated.
- **Station-based variables with proportional distribution**. [14] enforce a trip distribution proportional to the O-D trip demand as $x_{s_2}^{-,t_2} \leq \sum_{t=0}^{t_2-1} \sum_s x_s^{+,t} \frac{F_{s,s_2}^{t,t_2}}{f_s^{+,t}}$. Especially, the proportion is given by the ratio between the number of trips starting at station s in time-period t and ending at station s_2 in time-period t_2 (F_{s,s_2}^{t,t_2}) and the total number of trips starting at station s in time-period t ($f_s^{+,t}$). The authors also assume that the number of bikes returned during a specific time-period is not higher than the number of bikes rented in the previous periods multiplied by the corresponding proportion. Similarly, [33] consider a return ratio on the number of returns at station s divided by the total number of bikes currently used by customers during the time-period

t. The return demand at station s in period t is assumed to be the product of the return ratio and the total number of bikes being used.

- **O-D variables without distribution**. In this case, O-D variables will be created to represent the number of trips starting from one station and ending at another station during one particular time-period.
- **O-D variables with proportional distribution**. [9] and [27] apply a similar proportional distribution as [14]. The constraints $x_{s_1,s_2}^{t_1,t_2} \leq ab_{s_1}^{t_1} \frac{F_{s_1,s_2}^{t_1,t_2}}{f_{s_1}^{+,t_1}}$ imply that the trips rentals from a specific station are limited by the distribution ratio multiplied with the number of currently available bikes ($ab_{s_1}^{t_1}$).
- **Poisson distribution**. [34], [4], and [29] model the arrival of rentals and returns as a Poisson process, which implicitly models trip uncertainty.

These trip distribution constraints, as well as other alternatives, will be discussed in detail in Section 3.2.

2.4 Sequence of rebalancing, bike rental, and bike return events

In station-based BSSs, where the operator carries out station rebalancing using trucks, both customers and vehicles interact with the station inventories: customers may rent or return bikes, while trucks may drop off or pick up bikes. While in reality, customers and trucks interact with the station inventory at a specific moment in time, an optimization model aggregates these operations within each time-period.

Different assumptions can be made to deal with this issue. Some or all of these events can be assumed to happen simultaneously, allowing rentals and pick-ups to compensate for returns and drop-offs occurring within the same time-period. Such a generous assumption may achieve a high demand satisfaction within the optimization model. In practice, however, this may be overly optimistic and not perform well, i.e., rentals may occur at empty stations before returns and drop-offs, or returns may occur at full stations before rentals and pick-ups. Alternatively, one may assume that these events occur in a pre-defined chronological sequence within each time-period; for example, bike rentals occur first, then bike returns, and finally, the rebalancing operations. While such an assumption is more restrictive concerning demand satisfaction, it assumes that rentals can only be performed if sufficient bikes are available before the returns, and customer demands cannot benefit from the rebalancing operations that are assumed to happen at the end of the time-period.

Let us denote by **(r)** the event of vehicle **rebalancing**, **(a)** the event of customer **arrival** to return bikes, and **(d)** customer **departure**, i.e., bike rental. While models in the literature have assumed different event sequences, theoretically, any combination of these three event types is possible.

Table 1 summarizes the possible combinations, where events within the same parentheses are assumed to happen simultaneously. For example, (r)(a)(d) assumes that rebalancing is performed first, then customer arrivals, and then customer departures. In contrast (r+a+d) assumes that all three events happen at the same time. Note also that all sequences reported within the same row in Table 1 have the same order of events, but not necessarily within the same time-period. For example, both (r)(a)(d) and (a)(d)(r) assume that arrivals occur after rebalancing and departures occur after arrivals if the sequence is observed over several time-periods, e.g., (r)(a)(d)(r)(a)(d)(r)(a)(d), etc. However, such similarity does not guarantee a similar performance of the obtained solutions.

Within the existing literature, [10] use a series of station-visit events and extreme values of cumulated demand to discretize the planning horizon. Since the time required to pick up and drop off bikes is neglected, there is no particular order of events. However, their model essentially assumes (a+d)(r), which means that customers first rent and return bikes before vehicles rebalance. [31] assume that rebalancing happens first and then customers rent and return bikes. [13] use sequence (a)(r)(d), while [14] use sequence (a)(d+r). Several other works [e.g., 8, 9, 27, 35, 36, 37] ignore the issue of

Table 1: Potential sequences of events

Three separate events	(r)(a)(d) (r)(d)(a)	(a)(d)(r) (d)(r)(a) (d)(a)(r) (a)(r)(d)
Two events simultaneously	(r)(a+d) (a)(d+r) (d)(r+a)	(a+d)(r) (d+r)(a) (r+a)(d)
All events simultaneously	(r+a+d)	

event sequences, which is equivalent to assuming a simultaneous event sequence (r+a+d). Finally, a different approach is taken by [33], who subdivide each time-period into smaller time-segments and associate rental and return demand to such fine-grained time-segments. Bike pick-ups from rebalancing operations are assumed to happen at the first segment of each time-period, while drop-offs occur at the end. Such an assumption does not fit our classification scheme, but the discretization into time segments resembles the operating mode of our simulator.

Unfortunately, no studies are available exploring the degree of realism and effectiveness of the different event sequences. In Section 3.2.3, we will therefore explicitly review the modeling of the various alternatives and empirically evaluate their effectiveness.

Other assumptions and objective functions can be found in Appendix 1 of the Online Supplement.

3 Multi-period rebalancing modeling framework

We now present a general modeling framework that can be adapted to the various assumptions discussed in Section 2. To this end, we first propose a basic multi-period optimization model for dynamic rebalancing. We then formulate the different assumptions, which can be easily incorporated into the basic model.

3.1 Formulation of the basic model

We first consider a basic multi-period model with minimal assumptions. Its input parameters are summarized in Table 2. Namely, S denotes the set of stations, while V denotes the set of available vehicles. Each station $s \in S$ has a total of C_s docks, referred to as its capacity. Each vehicle $v \in V$ has a bike capacity $\hat{C}_v$. Parameters $D_{i,j}$ and $R^t_{s,s'}$ denote respectively the distance and transit time between stations i and j at time-period t. We consider a planning horizon with $|T|$ time-periods, where each time-period $t \in T$ has a duration of L_t minutes.

Table 2: Input parameters of the optimization model

Input	Definition
S	The set of stations.
V	The set of vehicles.
T	The set of discretized time-periods.
$D_{i,j}$	The distance between station $i \in S$ and $j \in S$.
C_s	The capacity of station $s \in S$.
$\hat{C}_v$	The capacity of vehicle $v \in V$.
L_t	The length (in minutes) of time-period $t \in T$.
d^1_s	The initial number of bikes in station $s \in S$.
$\hat{d}^1_v$	The initial number of bikes in vehicle $v \in V$.
$z^1_{s,v}$	1, if vehicle $v \in V$ is at station $s \in S$ at the beginning of planning; 0, otherwise.
$f^{+,t}_s$	The expected rental demand at station $s \in S$ in period $t \in T$.
$f^{-,t}_s$	The expected return demand at station $s \in S$ in period $t \in T$.
$R^t_{s,s'}$	Transit time for vehicles from station $s \in S$ to station $s' \in S$ in period $t \in T$.
$F^{t,t'}_{s,s'}$	The number of trips from station $s \in S$ at period $t \in T$ to $s' \in S$ at period $t' \in T$.

We formulate the rebalancing problem as a MILP and assume that each vehicle can only visit one station at each period.

Table 3: Decision variables of the optimization model

Variable	Definition
$\boldsymbol{r_{s,v}^{+,t}}$	The number of bikes picked up at station s by vehicle v in period t
$\boldsymbol{r_{s,v}^{-,t}}$	The number of bikes dropped off at station s by vehicle v in period t
$\boldsymbol{z_{s,v}^{t}}$	1, if vehicle $v \in V$ is at station $s \in S$ at period $t \in T$; 0, otherwise.
d_s^t	The number of bikes available in station $s \in S$ at the beginning of period t
$\hat{d}_v^t$	The number of bikes in vehicle $v \in V$ at the beginning of period t
$p_{s,s',v}^t$	1, if vehicle v is at station s in period t and at station s' in period $t+1$; 0, otherwise
$x_s^{+,t}$	The number of successful bike trips starting from station s in period t
$x_s^{-,t}$	The number of successful bike trips ending at station s in period t

The decision variables are summarized in Table 3. Variables $r_{s,v}^{+,t}/r_{s,v}^{-,t}$ represent the number of bikes picked up/dropped off at station s by vehicle v during period t. Variable $z_{s,v}^t$ takes value 1 if station s visited by vehicle v at period t; 0 otherwise. For each time-period, intermediate variables are used, such as the number of bikes available at stations/in vehicles, successful trips, and the routes of the vehicles. We employ rental and return demand without enforcing trip distribution, and use **station-based trip** variables $x_s^{+,t}$ and $x_s^{-,t}$.

Then, the basic MILP model reads as follows:

$$\min \quad \sum_{s\in S}\sum_{t\in T}(f_s^{+,t} - x_s^{+,t}) + \sum_{s\in S}\sum_{t\in T}(f_s^{-,t} - x_s^{-,t}) \tag{1}$$

$$\text{s.t.} \quad \hat{d}_v^{t+1} = \hat{d}_v^t + \sum_{s\in S}(r_{s,v}^{+,t} - r_{s,v}^{-,t}) \qquad \forall\, v \in V,\ t \in T \tag{2}$$

$$d_s^{t+1} = d_s^t - \sum_{v\in V}(r_{s,v}^{+,t} - r_{s,v}^{-,t}) - x_s^{+,t} + x_s^{-,t} \qquad \forall\, s \in S,\ t \in T \tag{3}$$

$$\sum_{s\in S} z_{s,v}^t = 1 \qquad \forall\, v \in V,\ t \in T \tag{4}$$

$$r_{s,v}^{+,t} + r_{s,v}^{-,t} \le \hat{C}_v z_{s,v}^t \qquad \forall\, s \in S,\ v \in V,\ t \in T \tag{5}$$

$$0 \le \hat{d}_v^t \le \hat{C}_v, 0 \le d_s^t \le C_s \qquad \forall\, s \in S,\ v \in V,\ t \in T \tag{6}$$

$$0 \le x_s^{+,t} \le f_s^{+,t}, 0 \le x_s^{-,t} \le f_s^{-,t} \qquad \forall\, s \in S,\ t \in T \tag{7}$$

$$0 \le r_{s,v}^{+,t}, r_{s,v}^{-,t} \le \hat{C}_v \qquad \forall\, s \in S,\ v \in V,\ t \in T \tag{8}$$

$$z_{s,v}^t \in \{0,1\} \qquad \forall\, s \in S,\ v \in V,\ t \in T \tag{9}$$

Objective function (1) minimizes the total lost rental and return demand in the planning horizon over all stations and time-periods. If required, it can be modified according to the preferences of the BSS operators (see Appendix 1.2 of the Online Supplement). Constraints (2) ensure that the number of bikes in each vehicle is synchronized with the vehicles' bike pick-ups and drop-offs. Constraints (3) manage the station inventory along time, considering the rebalancing operations and successful customer trips (i.e., rentals and returns). Note that we use the sequence (r+a+d) in our basic model. The initial inventory of stations d_s^1 is an input of the optimization model and can be obtained from the operators or through static rebalancing. Constraints (4) ensure that each vehicle is at exactly one station at each time-period, which forms the flow of vehicles sequentially. [13] use an alternative constraint: $\sum_{s'} p_{s,s',v}^t - \sum_{s'} p_{s',s,v}^{t-1} = 0$ ($\forall\, s,t,v$), which directly ensures that the flow out of station s for vehicle v at time-period t is equivalent to the flow of v into the station s at time $t-1$. Both of them indicate the relocation of vehicles along time. Note that, in our model, the vehicle can stay at the same station in the next time-period, i.e., $z_{s,v}^t = z_{s,v}^{t+1} = 1$. Constraints (5) ensure that a vehicle only operates at the station where it is currently located. Constraints (6) enforce that the number of bikes in each vehicle is limited by its capacity and the number of bikes at each station is within the station's capacity. Constraints (7) impose that the number of successful trips is bounded by the expected demand for rentals and returns. Constraints (8) force the number of picked-up/dropped-off bikes to respect the vehicle capacity.

The model can easily be reformulated using **O-D** variables $x_{s,s'}^{t,t'}$ instead of station-based variables. In this case, all occurrences of $x_s^{+,t}$ within (1)–(3) simply have to be replaced by $\sum_{s',t'} x_{s,s'}^{t,t'}$, while Constraints (7) have to be replaced by $x_{s,s'}^{t,t'} \leq F_{s,s'}^{t,t'}$.

3.2 Formulating different modeling assumptions

We now show how to extend the basic model to account for the additional modeling assumptions discussed in Section 2.

3.2.1 Time constraints.

The basic model assumes that vehicles can relocate to any other stations and carry out the rebalancing operations within the duration of a time-period. When the duration of the time-period is short, the resulting planning solution may become infeasible in practice. Since vehicles may not have sufficient time to relocate and rebalance bikes, time constraints (as discussed in Section 2.1) may be added to restrict the vehicle relocation between stations and rebalancing operations to the time available. We formulate time constraints as follows. First, for each pair of stations s and s', vehicle v, and time-period t, Constraints (10) enforce variable $p_{s,s',v}^t$ to take value 1 if both variables $z_{s,v}^t$ and $z_{s',v}^{t+1}$ are have value 1. Then, time constraints (11) guarantee that the transit time between stations and the operation time for picking up/dropping off bikes for each period will not surpass the available time L_t.

$$z_{s,v}^t + z_{s',v}^{t+1} - 1 \leq p_{s,s',v}^t \qquad \forall\ s, s' \in S,\ v \in V,\ t \in T \qquad (10)$$

$$\sum_{s \in S} \sum_{s' \in S} p_{s,s',v}^t R_{s,s'}^t + op \sum_{s \in S} (r_{s,v}^{+,t} + r_{s,v}^{-,t}) \leq L_t, \qquad \forall\ v \in V,\ t \in T \qquad (11)$$

$$p_{s,s',v}^t \in \{0,1\} \qquad \forall\ s, s' \in S,\ v \in V,\ t \in T, \qquad (12)$$

where op is the average operational time to pick up/drop off a single bike. Parameters $|T|$ and L_t can be altered by the decision-maker. In Section 5.3, we will test different lengths of time-periods along with the time constraints to explore their impact on the solution performance.

3.2.2 Trip distribution constraints.

We now discuss the proportionality distribution for station-based trip variables. The proportionality distribution constraints for O-D variables can be found in Appendix 2 of the Online Supplement.

Consider a trip starting at station s_1 in period t_1 and ending at station s_2 in period t_2. The station-based trip variables related to this trip are $x_{s_1}^{+,t_1}$ and $x_{s_2}^{-,t_2}$. The *proportional distribution* can be written as:

$$x_{s_2}^{-,t_2} \leq \sum_{t=0}^{t_2-1} \sum_{s \in S} x_s^{+,t} \frac{F_{s,s_2}^{t,t_2}}{f_s^{+,t}} \qquad \forall\ s_2 \in S,\ t_2 \in T \qquad \text{(TD1)}$$

$$x_{s_1}^{+,t_1} \leq \sum_{t=t_1+1}^{|T|} \sum_{s \in S} x_s^{-,t} \frac{F_{s_1,s}^{t_1,t}}{f_s^{-,t}}, \qquad \forall\ s_1 \in S,\ t_1 \in T \qquad \text{(TD2)}$$

where, as discussed, $\frac{F_{s,s_2}^{t,t_2}}{f_s^{+,t}}$ represents the proportion of all rental demand from s at time-period t that will be returned to s_2 at time-period t_2.

Constraints (TD1) impose that the number of bikes returned to station s_2 during period t_2 is no more than the bikes rented in the previous periods with a proportion of $\frac{F^{t,t_2}_{s,s_2}}{f^{+,t}_s}$. Conversely, constraints (TD2) impose that the number of bikes rented in station s_1 during period t_1 is no more than the bikes returned in the later periods with proportion $\frac{F^{t_1,t}_{s_1,s}}{f^{-,t}_s}$.

Similarly, instead of considering the number of rented/returned bikes, we can consider the number of available bikes/docks. To this end, we rewrite (TD1) and (TD2) to (TD3) and (TD4) by replacing $x^{+,t}_s$ with $ab^{t_1}_{s_1}$ and $x^{-,t}_s$ with $ad^{t_1}_{s_1}$.

$$x^{-,t_2}_{s_2} \leq \sum_{t_1=0}^{t_2-1} \sum_{s\in S} ab^t_s \frac{F^{t,t_2}_{s,s_2}}{f^{+,t}_s} \qquad \forall\ s_2 \in S,\ t_2 \in T \tag{TD3}$$

$$x^{+,t_1}_{s_1} \leq \sum_{t=t_1+1}^{|T|} \sum_{s\in S} ad^t_s \frac{F^{t_1,t}_{s_1,s}}{f^{-,t}_s}, \qquad \forall\ s_1 \in S,\ t_1 \in T \tag{TD4}$$

where ab^t_s and ad^t_s are the number of available bikes and docks respectively at station s in period t.

Given that station-based variables do not maintain the link between bike rental and return, we may use Constraints (TD5) below to enforce that the total number of rentals equals the total number of returns. When all trips are assumed to take no longer than one time-period, one may use Constraints (TD6) below to enforce a stronger relationship. Under the same assumption, Constraints (TD6) can also be derived by summing (TD2) all over s_1.

$$\sum_t \sum_s x^{+,t}_s = \sum_t \sum_s x^{-,t}_s, \tag{TD5}$$

$$\sum_s x^{+,t}_s = \sum_s x^{-,t+1}_s \qquad \forall t \in T. \tag{TD6}$$

Practical toy example. To develop an intuition of the impact of the trip distribution constraints, we consider the following toy example. We consider two time-periods with four stations, each of which has a pair of $[ab^t_s, ad^t_s]$ indicating the current number of bikes available for rentals and docks for returns, visualized in Figure 1. The value of ab^t_s is equal to $d^t_s - \sum_v (r^{+,t}_{s,v} - r^{-,t}_{s,v}) + x^{-,t}_s$ and it is analogous for the value of ad^t_s. The numbers circled in red along the arcs represent the trip demands for each station pair. Stations s_1 and s_2 may either have a sufficient (S) or insufficient (I) number of bikes to satisfy rental demand. Further, station s_3 and s_4 may either have a sufficient (S) or insufficient (I) number of empty docks to satisfy return demand. This leads to 4 different configurations shown in Figure 1.

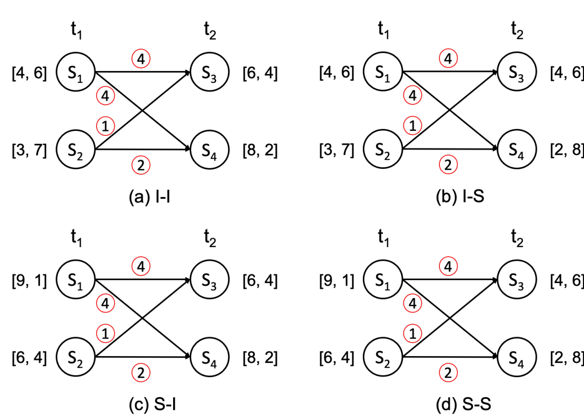

Figure 1: Toy example with 4 different situations of bike/dock availability

Ideally, for scenario I-I, a BSS operator would expect to see 4 trips to station s_3 and 2 trips to station s_4 to fill their empty docks, and will not mind whether the trips come from station s_1 or station s_2 as long as they have sufficient bikes. The ideal distribution is similar for scenario S-I. In contrast, in the case of scenario I-S, the operator would expect to see 4 trips from station s_1 and 3 trips from station s_2 such that all the available bikes can be used. However, there is no preference for the destinations of the 4 trips from station s_1. Clearly, scenario S-S is irrelevant since all trip demand is satisfied.

In order to compute the successful trips under different trip distribution constraints, we solved the basic model with each of them. For station-based trip variables, based on Constraints (3), (6), and (7), we have $x_{s_1}^{+,t_1} \leq \min\{f_{s_1}^{+,t_1}, ab_{s_1}^{t_1}\}$ and $x_{s_3}^{-,t_2} \leq \min\{f_{s_3}^{-,t_2}, ad_{s_3}^{t_2}\}$.

Minimizing the lost demand for cases I-I, I-S, and S-I in Figure 1 under the different trip distribution constraints results in the departures and arrivals $[x_{s_1}^{+,t_1}, x_{s_2}^{+,t_1}, x_{s_3}^{-,t_2}, x_{s_4}^{-,t_2}]$ indicated in Table 16. Here, all ideal trip distributions as expected by the BSS operator are indicated in the first row (an '*' refers to any coherent allocation). Note that all trip distribution constraints result in the same solution for case S-S since all trip demands can be satisfied. The solutions that are considered coherent with an ideal solution are indicated in bold.

Table 4: Trip distribution for 3 different demand scenarios under different trip distribution constraints for station-based variables

Constraints	I-I	I-S	S-I
Ideal solution	[*, *, 4, 2]	[4, 3, *, *]	[*, *, 4, 2]
without TD	[4, 3, 4, 2]	[4, 3, 5, 6]	[8, 3, 4, 2]
(TD1)	[4, 3, 3, 2]	**[4, 3, 3, 4]**	[8, 3, 4, 2]
(TD2)	[4, $\frac{22}{15}$, 4, 2]	[4, 3, 5,6]	**[$\frac{68}{15}$, $\frac{22}{15}$, 4, 2]**
(TD6)	[$x_{s_1}^{+,t_1} + x_{s_2}^{+,t_1} = 6$, 4, 2]	[4, 3,$x_{s_3}^{-,t_2} + x_{s_4}^{-,t_2} = 7$]	[$x_{s_1}^{+,t_1} + x_{s_2}^{+,t_1} = 6$, **4, 2]**
(TD1)+(TD2)	[$\frac{8}{3}$, 1, $\frac{5}{3}$, 2]	[4, $\frac{3}{2}$, $\frac{5}{2}$, 3]	[$\frac{8}{3}$, 1, $\frac{5}{3}$, 2]
(TD1)+(TD6)	[4, 0, 2, 2]	**[4, 3, 3, 4]**	[4, 0, 2, 2]
(TD2)+(TD6)	[4, $\frac{4}{3}$, $\frac{10}{3}$, 2]	[4, 2, 0, 6]	**[$\frac{68}{15}$, $\frac{22}{15}$, 4, 2]**
(TD3)	[4, 3, 3, 2]	**[4, 3, 3, 4]**	[8, 3, 4, 2]
(TD4)	[4, $\frac{22}{15}$, 4, 2]	[4, 3, 5, 6]	**[$\frac{68}{15}$, $\frac{22}{15}$, 4, 2]**
(TD3)+(TD4)	[4, $\frac{22}{15}$, 3, 2]	**[4, 3, 3, 4]**	**[$\frac{68}{15}$, $\frac{22}{15}$, 4, 2]**
(TD3)+(TD6)	[4, 1, 3, 2]	**[4, 3, 3, 4]**	[$x_{s_1}^{+,t_1} + x_{s_2}^{+,t_1} = 6$, **4, 2]**
(TD1)+(TD4)	[4, $\frac{22}{15}$, $\frac{112}{45}$, 2]	**[4, 3, 3, 4]**	[$\frac{68}{15}$, $\frac{22}{15}$, $\frac{124}{45}$, 2]
(TD2)+(TD3)	[$\frac{56}{15}$, $\frac{19}{15}$, 3, 2]	[4, $\frac{29}{15}$, 3, 4]	**[$\frac{68}{15}$, $\frac{22}{15}$, 4, 2]**

According to the observed trip distribution, (TD3)+(TD6) and (TD3)+(TD4) have the potential to produce trips that are close to an ideal solution. The combination of (TD1) and (TD2) will be tight for the feasible region, especially for full/empty stations. Constraints (TD1)+(TD6) introduce strict proportional limitations for rentals when the docks are insufficient for returns, which deviates from the ideal solution. Using only (TD1) may result in solutions with more rentals than returns.

Based on this analysis, within our empirical evaluation in Section 5, we will consider the station-based variable model without trip distribution constraints, with (TD1), (TD6), (TD3)+(TD6), and (TD3)+(TD4). The trip distribution of O-D variables, as well as the discussion, is illustrated in Table 16 of the Online Supplement.

3.2.3 Sequences of events

The basic station-based model (1)–(9) implicitly uses event sequence (r+a+d), where the three events are assumed to occur simultaneously. We now show how this model can be modified to take into account the different sequences of rebalancing, rental, and return events. We give several sequence examples which perform well in the following experiments or are used in the literature.

(r)(a)(d): Since the rebalancing operations performed by the vehicles occur at the beginning of each time-period, Constraints (13)–(16) need to be added, in order to ensure that arrivals consider the rebalancing and departures consider both rebalancing and departures.

$$\sum_{v} r_{s,v}^{+,t} \leq d_s^t \qquad \forall\ s \in S,\ t \in T \tag{13}$$

$$\sum_{v} r_{s,v}^{-,t} \leq C_s - d_s^t \qquad \forall\ s \in S,\ t \in T \tag{14}$$

$$x_s^{+,t} \leq d_s^t - \sum_{v} r_{s,v}^{+,t} + \sum_{v} r_{s,v}^{-,t} + x_s^{-,t} \qquad \forall\ s \in S,\ t \in T \tag{15}$$

$$x_s^{-,t} \leq C_s - d_s^t + \sum_{v} r_{s,v}^{+,t} - \sum_{v} r_{s,v}^{-,t} \qquad \forall\ s \in S,\ t \in T \tag{16}$$

(r)(d)(a): Since rebalancing occurs first, we require Constraints (13) and (14). In order for bike rentals to consider the previous rebalancing, we further require Constraints (17). Finally, bike returns are already correctly implemented due to Constraints (3) and (6).

$$x_s^{+,t} \leq d_s^t - \sum_{v} r_{s,v}^{+,t} + \sum_{v} r_{s,v}^{-,t} \qquad \forall\ s \in S,\ t \in T \tag{17}$$

(r)(a+d): Here, we use Constraints (13) and (14) since vehicles operate at the beginning of each period. The restrictions related to the rentals and returns are already satisfied by Constraints (3) and (6). The constraints required for all remaining event sequences can be found in Appendix 3 of the Online Supplement.

4 Dynamic rebalancing evaluation framework

Evaluation framework for rebalancing strategies. The framework used to obtain rebalancing strategies for given problem instances and to evaluate their estimated performance in practice is visualized in Figure 2. The input set contains the trip demand with exact time-stamps, referring, for example, to historical days with similar demand patterns (e.g., weekdays).

In order to analyze the effect of the various modeling assumptions in a controlled environment, we first generate synthetic problem instances, including a station network along with probability distributions of trip demand over a specified time horizon (in our case one day). For each problem instance, an input set and a test set are sampled, containing a certain amount of days of trip demands. The deterministic optimization model then receives as input the expected trip demand (i.e., a single demand scenario). While this point estimate may be provided by a predictive model, we here refrain from using a predictive model to focus on the impact of the modeling assumptions without potential interference from prediction errors. Instead, we compute the expected demand by averaging over the demand (for each time-period and station) of all days in the input set. Note that, if a stochastic (scenario-based) model was used, each scenario would contain the trip data of a different day.

The rebalancing strategies obtained from the optimization model are then applied to a simulated BSS, which aims at realistically estimating the performance (e.g., the lost demand) of the planning solutions on the various trip demand days within the test data set.

We also define an optimization-simulation-gap, representing the difference between the number of successful trips in the deterministic optimization model and the average number of successful trips simulated on each day in the input set. This gap therefore estimates the deviation from reality, largely explained by the temporal aggregation of the optimization model versus a FIFO policy that applies in reality.

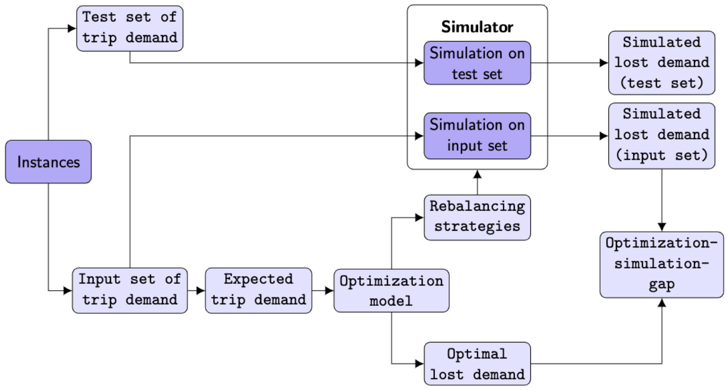

Figure 2: Evaluation framework for dynamic rebalancing strategies

Specifically, $\left(|\sum_{t,s} x_s^{+,t} - \bar{x}_s^{+,t}|\right) / \left(\sum_{t,s} \bar{x}_s^{+,t}\right)$ and $\left(|\sum_{t,s} x_s^{-,t} - \bar{x}_s^{-,t}|\right) / \left(\sum_{t,s} \bar{x}_s^{-,t}\right)$ compute the relative gaps of rentals and returns, respectively, over the entire planning horizon and all stations, where $x_s^{+,t}$/ $x_s^{-,t}$ are rentals/returns as computed by the optimization model and $\bar{x}_s^{+,t}$/ $\bar{x}_s^{-,t}$ are the average number of successful rentals/returns within the simulator.

Fine-grained simulator. Using a simulator to evaluate the performance of the proposed rebalancing strategies has been a common approach in the literature [13, 30]. Most simulators, however, aggregate the entire demand of each time-period, therefore allowing return demand to cancel out rental demand (and vice-versa). This is overly optimistic and deviates from the first-arrive-first-serve policy in practice. The required operating time for rebalancing is also typically ignored.

Aiming at a more realistic evaluation, we develop a discrete-event simulator taking into account more realistic operational BSS mechanisms. Each time-period used in the optimization model (spanning 30 or 60 minutes) is further discretized into 1-minute *time-slots* (which is sufficiently fine-grained to be considered real-time in practice). We consider both users' behaviors (rentals and returns) and trucks' operations (pick-ups and drop-offs) as discrete events, each of which is associated with a specific 1-minute time-slot. Rebalancing operations may occur in parallel to rentals and returns and depend on the time the truck arrives at the station. Customer trips and rebalancing operations are therefore considered in chronological order.

For ease of presentation, the operation mode of the simulator is now described verbally. First, the simulator *initializes* the station and vehicle inventories according to the input data. Each rental demand is associated with its respective 1-minute time-slot. Pick-up and drop-off attempts for the first time-period are associated with their respective time-slot, taking into consideration the time to load/unload bikes on/from the truck.

The simulator then scrolls through the time-slots minute by minute, attempting to *perform all scheduled events* for that time-slot. The operating rules in this iterative process can be summarized as follows: (i) A rental demand is satisfied if the station holds at least one bike. In this case, a return demand is created for the destination station and associated with a future time-slot based on an estimated travel time. If station inventory is insufficient, the rental demand is counted as lost. (ii) Analogously, a return demand is satisfied if a free dock is available. Otherwise, the bike is returned to the nearest station with an available dock. However, this return demand is then counted as lost. Note that lost returns can only occur if the corresponding rental demand was successful. (iii) Drop-off and pick-up attempts from rebalancing operations are carried out as best as the available inventory and available docks at the stations and the vehicles allow. (iv) Once a truck has finished the rebalancing attempt, it departs to the next station as prescribed by the planning solution for the next time-period. The arrival event is scheduled for a future time-slot based on the estimated travel time of the vehicle.

(v) Once a truck arrives at a new station, it is assumed to immediately start the rebalancing operations. However, rebalancing will not start before the first time-slot associated with the current time-period.

A pseudo-code of the simulator, along with a technical description can be found in Appendix 5 of the Online Supplement.

5 Computational experiments

We now report on different sets of computational experiments on both synthetic and real-world data to systematically explore the impact of the various modeling assumptions, based on the evaluation framework we proposed in Section 4.

In Section 5.1, we first elaborate on the synthetic problem instances used throughout the majority of our experiments. In Section 5.2, we compare dynamic and static rebalancing, as well as the usage of different variable types. Section 5.3 focuses on the impact of time discretization and time constraints. Section 5.4 analyzes the importance of the various trip distribution constraints. In Section 5.5, we cross-test whether the previous findings still hold when different variable domains are used. Section 5.6 focuses on the impact of the assumed event sequence. Finally, Section 5.7 empirically validates a variety of such modeling assumptions on real-world data from the Montreal bike-sharing system.

5.1 Generation of synthetic data and computing environment

Even though we have access to real-world trip data from different BSSs, the majority of our experiments are based on synthetically generated instances for a variety of reasons. First, the unobserved demand makes it difficult to obtain accurate rebalancing strategies and evaluate their performance. Second, existing data often contains errors and noise concerning trip and station inventory. Third, rebalancing operations carried out in the BSS alter station inventories, but existing data sets do not provide reliable data on the rebalancing carried out. We therefore develop an instance generator that aims at generating realistic instances with BSS networks of different sizes and characteristics, as well as trip data that is coherent with trips observed in reality (see details in Appendix 4 of the Online Supplement). Note, however, that we validate the most relevant conclusions within a case study based on real-world data in Section 5.7.

For the purpose of our study, we only focus on weekdays, since they have similar demand patterns for work-related trips and demand tends to be much higher than on weekends. We divide the entire daily trip demand into four types. People who live outside city centers and work inside city centers typically use similar origin (outside city centers) and destination stations (within city centers) during peak hours. These trips are denoted as OI trips. Trips of people who live and work outside the centers are referred to as OO trips. In contrast to work-related OI and OO trips, RD trips refer to random trips occurring during the day and RN trips refer to random trips during the night. Such random trips do not have the same origin and destination stations. The departure time for each trip type is characterized by a Beta distribution (see Appendix 4.2 of the Online Supplement). The average demand per 30-minute duration for weekdays of one week in July 2019 at BIXI is visualized in Figure 3(a). For comparison, the trip demand averaged over 500 days as generated by our instance generator is displayed in Figure 3(b) and shows a similar pattern as the trip demand observed at BIXI.

We generate 3 ground truths with different station networks. For each ground truth, we generate 5 instances with the proportions for the four trip types. For each instance, we generate 5000 weekdays, from which a single expected demand is computed for each time-period and station as input for the optimization model. We then generate 1000 weekdays of trips as test data, on each of which the planned solution will be simulated. Table 5 shows the characteristics of the 3 ground truths. The percentage of stations within the city centers and those associated with each trip type are indicated in rows 2 to 5. Ground Truth 3 (GT3) has more work-related trips, compared to Ground Truth 1

(GT1). Ground Truth 2 (GT2) has two city centers under the same trip pattern as GT1. A detailed description can be found in Appendix 4.3 of the Online Supplement.

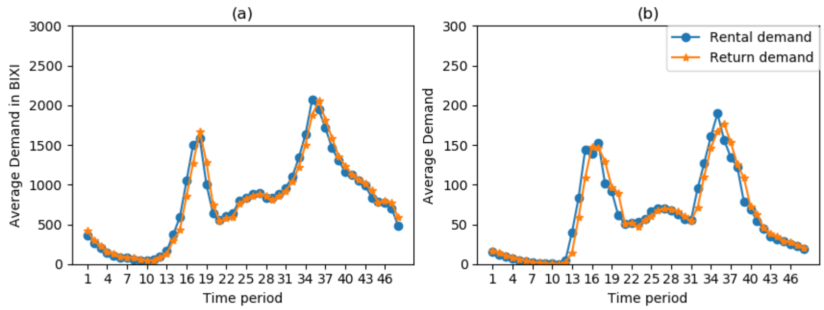

Figure 3: Total rental and return demand over 24 hours (48 time-periods) : (a) Average weekday demand at BIXI ; (b) Average demand in the synthetic instance GT1 (Sample of 500 days)

.

Table 5: The parameters for the ground truths

		GT1	GT2	GT3
Station Network	Number of city centers	1	2	1
	City center capacity	26%	35%	26%
Trip Pattern	OI	32% - $\beta(3,8)$	32% - $\beta(3,8)$	55% - $\beta(3,8)$
	OO	32% - $\beta(3,7)$	32% - $\beta(3,7)$	25% - $\beta(3,7)$
	RD	23% - $\beta(3,7)$	23% - $\beta(3,7)$	15% - $\beta(3,7)$
	RN	13% - $\beta(6,8)$	13% - $\beta(6,8)$	5% - $\beta(6,8)$

Our optimization models are solved by IBM ILOG CPLEX on 2.70 GHz Intel Xeon Gold 6258R machines with 8 cores. The stopping criterion for the optimization model is a MIP optimality gap of 0.01% and a maximum running time of 24 hours. In the following, we will report results for GT1 and GT2. While GT3 has been harder to solve, its results demonstrated the same conclusions. Detailed results can be found in Appendix 6 of the Online Supplement.

5.2 Impact of initial station inventory and trip variable types

To quantify the impacts of the initial inventory at stations, we define two baselines as the pre-allocated inventory.

- **Baseline 1: Inventory proportional to rental demands without rebalancing**. The initial inventories of stations at the beginning of a day are set to predefined levels proportional to the rental demands of the first time-period in the planning horizon. We round the values to the closest integer, respecting the station capacities and the total number of bikes available in the system.
- **Baseline 2: Static rebalancing only**. The optimal static rebalancing is obtained by solving the problem given by (30)–(34) in Appendix 6.2 of Online Supplement, where the inventories for the first time-period are decision variables that sum to the total number of available bikes in the system.

In the following, we set the initial inventory of the stations according to these two baselines in the optimization model and run our simulator without dynamic rebalancing for all 3 ground truths. We consider a planning horizon from 6 a.m. to 1 p.m. (i.e., 7 hours) and divide the planning horizon into 14 time-periods with a length of 30 minutes each. We calculate the average rental and return demands for each instance over the input set at each station of each time-period. Rebalancing strategies are obtained through the optimization model and then applied in the simulator to estimate the lost demand on the test set.

Table 6 summarizes the results for GT1 and GT2. The results for GT3, as well as for the O-D variables can be found in Appendix 6.2 of the Online Supplement. We report the optimal value of the objective function as 'O.F. Value' and the running time of the optimization model as 'Opt. Time' in minutes. The 'MIP gap' refers to the optimality gap as reported by CPLEX when the stopping criterion is reached. The lost demand is computed as the relative gap between successful trips and the original demand specified in the problem instances. To be specific, $\frac{\sum_{s,t}(f_s^{+,t}-\hat{x}_s^{+,t})}{\sum_{s,t} f_s^{+,t}}$ defines the lost rental demand over the entire planning horizon, where $\hat{x}_s^{+,t}$ is the number of successful rentals in simulator. Similarly, the lost return demand is computed as $\frac{\sum_{s,t}(\hat{x}_s^{+,t}-\hat{x}_s^{-,t})}{\sum_{s,t} \hat{x}_s^{+,t}}$, where $\hat{x}_s^{-,t}$ is the number of successful returns in simulator. Since, in practice, return demand does not exist when the corresponding rental demand is unsuccessful, the lost returns are only associated with successful rentals $\hat{x}_s^{+,t}$. Lost return demand has to be interpreted critically since the relative lost return may be high when the lost rental is low (which doesn't indicate a low-quality planning solution). In our result analysis, we, therefore, emphasis on the lost rental.

For each ground truth, the initial inventory observed from Baselines 1 and 2 is directly applied to the simulator without any rebalancing operations, whose average lost demand over 5 instances is reported as 'Baseline 1/2 without rebal.'.

Table 6: Station-based model with baseline 1 and baseline 2 (60 stations, 4 trucks, 30 mins)

	Baselines, Configuration, Trip Modeling	O.F. Value	Opt. Time (mins)	MIP Gap (%)	Lost Demand (%)	
					Rental	Return
GT1	Baseline 1 without rebal.	-	-	-	26.05	10.27
	Baseline 2 without rebal. (static)	-	-	-	10.40	11.79
	Baseline 1 dyn.rebal. station-based	44.2	1440	0.10	11.11	2.12
	Baseline 2 dyn.rebal. station-based	0.8	< 1	0.00	8.78	7.99
GT2	Baseline 1 without rebal.	-	-	-	21.22	6.85
	Baseline 2 without rebal. (static)	-	-	-	11.61	3.18
	Baseline 1 dyn.rebal. station-based	6.4	294	0.01	11.22	1.78
	Baseline 2 dyn.rebal. station-based	0.5	< 1	0.00	9.46	1.78

According to the first two rows for each GT in Table 6 (baselines without rebalancing), the initial station inventory seems important to the performance of the BSS. Compared to Baseline 1, static rebalancing (Baseline 2) can significantly improve the lost demand. However, static rebalancing is still insufficient to meet customer demand.

Based on the initial station inventory from Baselines 1 and 2, rows 3-4 for each GT compare the impact of the two different strategies with additional dynamic rebalancing. For the dynamic rebalancing, we use model (1)–(9) without trip distribution and time constraints.

The lost rental is improved when dynamic rebalancing is applied. The performance of dynamic rebalancing varies substantially between the two baselines, which highlights the importance of optimizing the initial station inventory before the dynamic rebalancing. For GT1, the lost rental for dynamic rebalancing with Baseline 1 is higher, leading to a decrease in actual return demands. Given that we only consider lost returns for successful rentals, it is possible that the relative lost return is small when the relative lost rental is high (i.e., only a few successful rentals).

Since strategies based on Baseline 2 outperform those based on Baseline 1, in the following experiments, we will use Baseline 2 to define the initial inventory of each station. Note that we use pre-defined initial locations and inventories for trucks because previous experiments have shown that such assumptions do not affect the performance (see Appendix 6.1 of Online Supplement for details).

5.3 Impact of time discretization and time constraints

We now explore the impact of the length of time-periods and the use of time constraints. We consider two lengths of time-periods: 30-minute and 60-minute, and with or without time constraints. The comparative results for station-based trip variables are summarized in Table 24. The results of similar experiments using O-D variables can be found in Appendix 6.3 of the Online Supplement.

Table 7: Station-based model with/without time constraints in 30/60 mins (60 stations, 4 trucks)

	Time Period (mins)	Time Constraints	O.F. Value	Time (mins)	MIP Gap (%)	Lost Demand (%)		Opt-sim-gap (%)	
						Rental	Return	Rental	Return
GT1	30	No	0.8	<1	0.00	8.78	7.99	9.62	15.88
	30	Yes	0.9	69	0.00	8.13	6.06	8.84	12.69
	60	No	9.5	<1	0.00	8.50	4.39	8.52	11.24
	60	Yes	9.5	2	0.00	8.73	5.85	8.80	13.27
GT2	30	No	0.5	<1	0.00	9.46	1.78	10.42	9.21
	30	Yes	0.5	28	0.00	8.48	1.90	9.24	8.18
	60	No	10.4	<1	0.00	9.72	1.20	9.88	8.90
	60	Yes	10.5	1	0.00	9.63	1.40	9.76	9.02

As shown in Table 24, 30-minute time-periods allow for more rebalancing operations within the optimization model, leading to smaller optimal objective function values than those for cases with 60-minute time-periods. Note again, that in our simulator, we postpone the rebalancing operations if the truck cannot reach the station in time due to long relocation distances. Using 30-minute time-periods, the lost rental without time constraints may therefore be worse than in the case of 60-minute time-periods. Coherently, time constraints with 30-minute time-periods give the best overall performance. Thus, if we have a tolerance for optimization time and the distances between stations tend to be large, a short time-period (30 mins) with time constraints seems beneficial. For longer time-periods, time constraints do not seem necessary. Using the model with O-D variables, we reach similar conclusions (see Appendix 6.3 of the Online Supplement).

5.4 Impact of trip distribution constraints

We implement optimization models with various TD constraints as discussed in Section 3.2 for station-based trip variables with Baseline 2. The results of station-based trip variables are shown in Table 8. The results of O-D variables can be found in Table 28 of the Online Supplement.

Table 8: Station-based model with different trip distribution constraints (60 stations, 4 trucks, 30mins)

	Constraints	O.F. Value	Time (mins)	MIP Gap (%)	Lost Demand (%)		Opt-sim-gap (%)	
					Rental	Return	Rental	Return
GT1	(TD1)	600.3	1440	1.41	3.65	6.34	0.56	40.07
	(TD6)	137.7	<1	0.00	9.94	11.22	5.89	14.28
	(TD3)+(TD6)	240.3	<1	0.00	8.61	6.56	0.01	2.35
	(TD3)+(TD4)	170.5	1440	0.13	6.15	1.57	0.11	1.82
	None	0.8	<1	0.00	8.78	7.99	9.62	15.88
GT2	(TD1)	588.6	1440	1.48	4.78	1.76	1.20	40.70
	(TD6)	101.3	<1	0.00	10.86	3.21	8.40	7.45
	(TD3)+(TD6)	165.0	<1	0.00	8.68	2.08	3.14	0.94
	(TD3)+(TD4)	150.0	1440	0.06	7.34	1.20	2.05	0.56
	None	0.5	<1	0.00	9.46	1.78	10.42	9.21

According to Table 8, the use of Constraints (TD1) performs best in terms of lost rental for both input and test sets even if optimality has not been proven within 24h. This suggests that (TD1) reflects the flow of rentals more realistically as also shown by the low rental Opt-sim-gap. Since (TD1) imposes a strict restriction for returns, the lost return of the optimization model is quite high, and lost rental is low. That leads to a small opt-sim-gap for rental but a large one in return. Constraints (TD3)+(TD4) also provide a good performance. Both sets of Constraints (TD1) and (TD3)+(TD4) include the proportionality characteristics of the trip flow. They improve performance, but they require longer computing times.

Based on these experimental results, in the following, we will restrict our analysis to station-based variables with (TD1), (TD3)+(TD4), and (None).

5.5 Impact of variable domains

If the variable domains were selected such that they represent more realistically the BSS operations, trip variables would be binary, while station and vehicle inventory, as well as the rebalancing variables, would be integer. However, using such variable domains may result in models that may be difficult to solve and restricted by certain trip distribution constraints, severely underestimating successful trips. We now explore the impact of using different variable domains. In our models, routing variables $z^t_{s,t}$ are always binary. Let variables d^t_v, $r^{+,t}_{s,v}$, and $r^{-,t}_{s,v}$ be referred to as rebalancing variables and variables d^t_s, $x^{+,t}_s$, and $x^{-,t}_s$ be referred to as station variables. In the previous experiments, both rebalancing and station variables have been continuous, which we denote as an *All-continuous* model. Here, we also consider the other two cases: the *All-integer* model and the *Partially-integer* model. In the All-integer model, both rebalancing and station variables are integers. In the Partially-integer model, the station variables are continuous, while the rebalancing variables are integers.

Table 9: Station-based model with different variable domains and trip distribution constraints for GT1 (30 stations, 2 trucks, 30 mins)

Variable Domains	Constraints	O.F. Value	Time (mins)	MIP Gap (%)	Lost Demand (%)		Opt-sim-gap (%)	
					Rental	Return	Rental	Return
All-continuous	(TD1)	262.7	1440	0.18	2.93	1.86	0.82	43.73
	(TD3)+(TD4)	82.7	19	0.00	6.00	0.87	1.62	2.86
	None	0.5	< 1	0.00	8.95	4.73	9.83	11.80
Partially-integer	(TD1)	263.6	1440	0.33	2.95	1.29	0.75	44.11
	(TD3)+(TD4)	82.8	50	0.00	5.78	1.47	1.96	2.32
	None	0.5	< 1	0.00	6.72	3.56	7.20	7.81
All-integer	(TD1)	656.5	140	0.08	3.89	4.62	30.43	81.39
	(TD3)+(TD4)	408.3	< 1	0.00	9.22	1.96	29.56	29.60
	None	380.5	< 1	0.00	10.50	7.65	24.88	22.36

Since the optimization models with 60 stations and trip distribution constraints (TD1) and (TD3)+(TD4) cannot be solved within the given time limit, we also carry out experiments with the 30-station network to reliably explore the impact of such constraints coupled with different variable domains. Specifically, we consider two ground truths with 30 stations using the same configurations as GT1 (see Table 5).

The results of the corresponding experiments for GT1 under different trip distribution constraints are summarized in Table 9. Surprisingly, the All-integer model is more tractable. Although we have not conducted a complete analysis of this behavior, we observed that more cuts are generated by CPLEX for the All-integer model, which helps in improving the dual bound.More precisely, under Constraints (TD1), three of the five instances of the All-integer model are solved to optimality within 24 hours. For the All-continuous and Partially-integer models under Constraints (TD1), none of the instances has been solved to optimality within the time limit. However, the MIP gaps are relatively small. As previously concluded from Table 8, we again observe that Constraints (TD1) provide the lowest lost demand, and this is consistent for all types of variable domains. In terms of the performance for different variable domains, even though it is fast to solve, the All-integer model provides the worst performance, which indicates that restricting station variables to integer values may result in too conservative trips. The Partially-integer model introduces an improvement and performs best in most of the cases, especially with constraints (TD3)+(TD4) and (None). The results for O-D variables can be found in Appendix 6.4 of the Online Supplement.

5.6 Impact of event sequences

Recall that using no particular event sequence, which equals (r+a+d), within the All-continuous station-based model without trip distribution constraints yields an average rental loss of 8.78% on the test set of the 60-station network instances (see Table 8). Based on the model (1)–(9) with continuous variables and the event sequences reviewed in Section 3.2.3, we now analyze the impact of such event sequences for problem instances on the 60-station network in Table 10.

Table 10: Station-based model with different sequences of events (60 stations, 4 trucks, 30 mins)

	Sequences of events	O.F. Value	Time (mins)	MIP Gap (%)	Lost Demand(%)		Opt-sim-gap(%)	
					Rental	Return	Rental	Return
GT1	(r)(a)(d)	1.0	288	0.00	7.91	5.22	8.62	1.39
	(a)(d)(r)	3.3	298	0.00	7.64	4.63	8.19	11.83
	(d)(r)(a)	13.4	1399	0.04	6.02	5.81	5.38	9.89
	(r)(d)(a)	8.2	1013	0.03	5.61	3.26	5.36	6.53
	(d)(a)(r)	15.1	879	0.01	6.40	3.48	5.70	7.65
	(a)(r)(d)	2.0	< 1	0.00	7.51	2.79	8.12	8.04
	(r)(a+d)	0.8	< 1	0.00	8.49	6.86	9.28	14.12
	(a+d)(r)	1.6	< 1	0.00	7.42	6.92	7.98	12.87
GT2	(r)(a)(d)	3.2	288	0.00	8.05	3.61	8.71	9.35
	(a)(d)(r)	1.9	864	0.49	8.33	5.11	8.98	11.55
	(d)(r)(a)	17.5	1440	0.16	6.75	5.51	5.88	10.14
	(r)(d)(a)	11.9	1440	0.08	6.39	3.79	5.93	7.77
	(d)(a)(r)	20.0	1341	0.13	6.38	3.77	5.26	7.74
	(a)(r)(d)	3.8	< 1	0.00	7.87	1.93	8.50	7.25
	(r)(a+d)	0.7	< 1	0.00	8.40	3.29	9.15	9.63
	(a+d)(r)	1.9	576	0.03	7.98	3.66	8.53	9.56

Sequences (d)(r)(a), (r)(d)(a), and (d)(a)(r) perform best for lost rental and return with a small opt-sim-gap. Although GT3 instances are hard to solve, these three sequences still perform well (see Table 30 in Appendix 6.5 of Online Supplement). In contrast, the sequences used in the literature (i.e., (r+a+d), (a)(r)(d), and (a+d)(r)) have performed less well in our experiments. Instead, sequences (d)(r)(a), (r)(d)(a), and (d)(a)(r) may be a better choice, reducing lost rental by an additional 1%-2%.

Table 11 shows the results of the same experiments for the problem instances on a 30-station network and 60-minute time-periods. Here, all instances have been solved to optimality and lead to the same conclusions. Particular sequences help reduce the lost rental: sequences (d)(r)(a) and (r)(d)(a) reduce the lost rental to around 7.45% from 9.12% in the test set.

In Section 5.4, we have concluded that it is beneficial to use trip distribution constraints (TD1) when no particular event sequence is used. The results discussed above suggest that it is beneficial to use a specific event sequence, such as (r)(d)(a) when no trip distribution constraints are used.

We now explore the combination of those two modeling assumptions and further backtest on different variable domains. To this end, Table 12 summarizes the results for GT1 problem instances on the 30-station network using trip distribution constraints (TD1) and various event sequences for All-continuous and Partially-integer variable domains. The models for both variable domains seem to perform similarly well in terms of lost rentals, while the All-continuous models tend to be solved faster.

When comparing with the results in Table 11, the introduction of Constraints (TD1) further decreases the lost rental for any of the event sequences. However, the improvement for (r+a+d) is the highest, indicating that using Constraints (TD1) without any specific event sequence may be the best option if the longer computing time is acceptable. Operators may therefore opt for event sequence (r)(d)(a) with all-continuous variables without trip distribution constraints if a quick solution is required, or Constraints (TD1) without a specific event sequence if higher computing times can be tolerated.

Table 11: Station-based model with different sequences of events (30 stations, 2 trucks, 60 mins)

	Sequences of events	O.F. Value	Time (mins)	MIP Gap (%)	Lost Demand (%)		Opt-sim-gap (%)	
					Rental	Return	Rental	Return
GT1	(r)(a)(d)	10.4	< 1	0.00	8.95	2.09	8.83	7.94
	(a)(d)(r)	15.0	< 1	0.00	9.38	1.37	8.47	7.63
	(d)(r)(a)	37.7	< 1	0.00	7.43	1.23	1.22	6.01
	(r)(d)(a)	33.1	< 1	0.00	7.45	1.10	1.93	6.10
	(d)(a)(r)	42.1	< 1	0.00	7.89	1.81	1.04	7.00
	(a)(r)(d)	11.3	< 1	0.00	8.62	1.92	8.29	7.34
	(r)(a+d)	5.3	< 1	0.00	9.81	2.08	9.89	9.94
	(a+d)(r)	9.8	< 1	0.00	9.53	2.48	8.72	10.05
	(r+a+d)	5.3	< 1	0.00	9.12	2.55	9.07	9.76

Table 12: Station-based model with (TD1) and sequences of events (30 stations, 2 trucks, 60 mins, GT1)

	Sequences of events	O.F. Value	Time (mins)	MIP Gap (%)	Lost Demand (%)		Opt-sim-gap (%)	
					Rental	Return	Rental	Return
All-continuous	(r)(d)(a)	499.4	22.9	0.00	6.62	4.09	14.36	65.34
	(d)(a)(r)	515.1	7	0.00	6.78	5.66	16.50	65.35
	(a+d)(r)	503.4	6	0.00	7.10	5.14	14.33	65.12
	(r+a+d)	485.5	237.16	0.00	6.32	3.87	12.37	65.22
Partially-integer	(r)(d)(a)	500.1	18	0.00	6.72	3.22	14.41	65.62
	(d)(a)(r)	515.6	5	0.00	6.73	6.32	16.69	65.08
	(a+d)(r)	504.1	7	0.00	6.97	6.59	14.57	64.65
	(r+a+d)	485.9	440.67	0.00	6.64	3.74	12.15	65.13

5.7 Case study on BIXI Montreal data

We now validate our previous findings by means of a case study based on real-world data from BIXI Montreal [38].

Data preprocessing. We focus on the 2019 season. To ensure a coherent association of historical arrivals and rentals to the given station IDs, we first discarded stations that changed locations throughout the 2019 seasons by more than 1 km (which is common given that the operator relocates stations due to events, construction, or holidays). The resulting network contained 606 stations, which, obviously, would result in optimization models that are too large to solve directly. Given that vehicle relocation is time-consuming, efficient rebalancing solutions typically relocate locally rather than over large distances. It is therefore reasonable to assume that the network can be subdivided into sub-clusters [see, e.g., 35, 39, 40, 41, 42, 43, etc.].

To this end, we first cluster stations via k-means based on demand pattern similarity [see, e.g., 35, 40, 42], ensuring the inclusion of city center stations and work-related trips. Similar to [39], we also limit the maximal distance between stations that belong to the cluster, which is motivated by the fact that the vehicles only travel within one cluster. We selected a cluster with 53 stations distributed around the downtown and Plateau areas (see Figure 4), which has approximately the same number of total rentals and returns [28] (a requirement, which is obviously satisfied in closed BSSs systems).

We focus on days with high demand, i.e., weekdays within the summer season of 2019, which have therefore not been affected by the COVID-19 pandemic. Outlier days with extremely low numbers of trips (e.g., due to bad weather or special events) are excluded, resulting in a total of 50 days. Aligned with the experiments on synthetic problem instances, we here consider a planning horizon from 7 a.m. to 2 p.m. (7 hours) with 14 time-periods, each of which spans 30 mins.

Figure 4: Considered station cluster from BIXI Montreal (generated through Google Maps [44])

Empirical results. With the objective of validating the most relevant conclusions drawn from the experiments on synthetic problem instances, we here use the station-based model. As opposed to O-D variables, the use of station-based rental and return variables $x_s^{+,t}$ and $x_s^{-,t}$ also enables us to aim at capturing trips to and from all stations in the original network, not only those included in the considered cluster. We further focus on the impact of variable domains, the event sequences, and the best-performing trip distribution constraints (TD1). Note that the use of time constraints is not necessary, given that all stations within the clusters are located sufficiently close to each other.

Trip constraints (TD1) consider the proportion of successful rentals that depart from any station to a specific station. Given that our model only uses stations within the considered cluster $C \subset S$, this proportion cannot be computed. We may rewrite the right-hand side of (TD1) as $\sum_{t=0}^{t_2-1}\sum_{s\in C} x_s^{+,t}\frac{F_{s,s_2}^{t,t_2}}{f_s^{+,t}} + \sum_{t=0}^{t_2-1}\sum_{s\in S\setminus C} x_s^{+,t}\frac{F_{s,s_2}^{t,t_2}}{f_s^{+,t}}$, where the first term accounts for successful trips from within the cluster, and the second term refers to rentals from stations outside the considered cluster. We replace $x_s^{+,t}(s \in S\setminus C)$ by $f_s^{+,t}(s \in S\setminus C)$ within the second term, therefore optimistically assuming that rentals outside the cluster are all satisfied. The adapted trip constraint then writes as follows:

$$x_{s_2}^{-,t_2} \leq \sum_{t=0}^{t_2-1}\sum_{s\in C} x_s^{+,t}\frac{F_{s,s_2}^{t,t_2}}{f_s^{+,t}} + \sum_{t=0}^{t_2-1}\sum_{s\in S\setminus C} F_{s,s_2}^{t,t_2} \qquad \forall\ s_2 \in C,\ t_2 \in T.$$

Table 13 summarizes the results of the corresponding experiments. Here, Baseline 2 refers to the model that only carries out (overnight) static rebalancing (see Section 5.2). In contrast to the results on synthetic problem instances, the improvement through dynamic rebalancing (as opposed to Baseline 2 only) is not impressive when no trip distribution constraints are used. The small improvement may be explained by the fact that the here-considered real-world data only contains successful trips. All other observations are well aligned with the results for synthetic instances. Using partially-integer variable domains without trip distribution constraints and without a specific event sequence reduces the estimated lost demand (compare Table 9). Further, the use of adapted trip distribution constraints

(TD1) positively impacts lost demand (compare Table 9, and Tables 11 and 12). Interestingly, the combination of using partially-integer domains with trip constraints seems particularly effective on real-world data, providing the lowest lost demand among all configurations. Finally, event sequences (r)(d)(a) and (d)(a)(r), which have been found to be among the best-performing sequences on synthetic instances, here also perform well with both all-continuous variables and partially integer variables in lost rental and return without trip distribution constraints. Finally, we note that all models have been solved to optimality within 1 minute, which is, of course, sufficiently fast for use in practice.

Table 13: Station-based model (53 stations, 4 trucks, 30 mins, 50 days high demand, Baseline 2)

	Sequences of events	(TD1)	O.F. Value	Time (mins)	MIP Gap (%)	Lost Demand (%)		Opt-sim-gap (%)	
						Rental	Return	Rental	Return
Baseline 2	-	-	-	-	-	8.32	6.12	-	-
All-continuous	(r)(d)(a)		0.0	< 1	0.00	7.43	6.15	20.95	18.04
	(d)(a)(r)		0.0	< 1	0.00	7.19	6.13	20.93	18.07
	(a+d)(r)		0.0	< 1	0.00	8.32	6.11	20.75	18.05
	(r+a+d)		0.0	< 1	0.00	9.42	7.65	20.82	18.30
Partially-integer	(r)(d)(a)		0.0	< 1	0.00	7.54	7.22	20.58	18.55
	(d)(a)(r)		0.0	< 1	0.00	7.36	6.14	20.92	18.05
	(a+d)(r)		0.0	< 1	0.00	8.32	6.12	20.75	18.05
	(r+a+d)		0.0	< 1	0.00	8.32	6.12	20.75	18.05
All-continuous	(r)(d)(a)	✓	453.4	< 1	0.00	5.70	6.67	20.98	25.50
	(d)(a)(r)	✓	453.4	< 1	0.00	3.88	7.21	20.83	25.29
	(a+d)(r)	✓	453.4	< 1	0.00	5.09	4.99	20.66	25.36
	(r+a+d)	✓	453.4	< 1	0.00	6.31	6.23	20.84	25.29
Partially-integer	(r)(d)(a)	✓	453.5	< 1	0.00	4.53	6.96	20.91	25.27
	(d)(a)(r)	✓	453.5	< 1	0.00	2.82	6.59	20.60	25.42
	(a+d)(r)	✓	453.4	< 1	0.00	2.89	6.34	20.61	25.25
	(r+a+d)	✓	453.4	< 1	0.00	5.79	6.07	20.70	25.36

6 Conclusions

In this paper, we aimed to disentangle and structure the various modeling assumptions and constraints used in the literature on Mixed-Integer Programming models for BSSs rebalancing optimization. To this end, we first surveyed the literature according to modeling techniques and assumptions, with a particular focus on multi-period models. We then introduced a modeling framework, rooted in a basic model, and showed how to adapt it to the various modeling assumptions. Finally, we evaluated the performance of the planned solutions induced by the different model variants as realistically as possible. Specifically, we generated different ground truths that propose BSSs networks and trip patterns matching observed trip patterns at BIXI Montreal. We then developed a fine-grained discrete-event simulator for truck movement, rebalancing operations, as well as bike rentals and returns on a minute-to-minute basis.

6.1 Summary recommendation

Based on the simulation results on a large set of test instances, we focused on two performance measures to analyze the appropriateness of the various modeling assumptions: the lost rental and return demand observed throughout the simulation, and the simulation-optimization-gap that indicates the deviation between the lost demand observed as estimated by the optimization model and by the simulator. Extensive numerical experiments on problem instances with networks including 30 and 60 stations and three different ground truths were carried out. Experiments have also been carried out on real-world data with 53 stations from the BIXI Montreal 2019 summer season. While one is required to be more careful when drawing conclusions from such results, given that the observed trip data refers to trips satisfied in the past, the corresponding conclusions tend to align. Based on these results, our principal conclusions can be summarized as follows:

(i) On the synthetic data sets, adding dynamic rebalancing to static rebalancing reduces the lost rental demand by an additional 2% (e.g., from 10.43% to 8.79% in Table 6).

(ii) Using station-based trip variables instead of more detailed trip variables based on origin-destination pairs generally appears to be competitive and results in faster solution times.

(iii) Shorter time-periods tend to allow for planning more rebalancing operations but may require time constraints to ensure that the resulting rebalancing is time-feasible in practice.

(iv) Trip distribution constraints, especially (TD1), reflect more realistically the trip flow observed in practice and may strongly improve the lost demand on both synthetic and real-world data (e.g., from 8.78% to 3.65% in Table 8; from 9.42% to 6.31% in Table 13); further, the best performing trip distribution constraint(s) are not necessarily those used in the literature.

(v) Using integer variables exclusively for truck routes, while keeping all other variables continuous (even the pick-up and drop-off decisions in the rebalancing operations) generally approximates reality sufficiently well; in some specific cases, it is beneficial to impose integrality on the rebalancing variables, while it does not seem beneficial to use integer variables for all decisions that, in reality, would also be integer.

(vi) Exploring the various sequences in which bike rentals, bike returns, and rebalancing operations may occur yields interesting conclusions. In particular, event sequences that have not been studied in the literature perform particularly well and tend to reduce lost rental by an additional 2% - 3% (Tables 10 and 11). Coupling specific event sequences, in particular (d)(a)(r), with our proposed trip distribution constraints (TD1) provides consistently low lost demand in short computing times on both synthetic and real-world instances.

The optimization model for the real-world problem instance has been solved within 1 minute of computing time, while the solution time for synthetic instances may vary strongly with the model configuration. When computing resources are limited and quick decisions are required, using a combination of trip distribution constraints with one of the newly proposed event sequences, in particular the above-mentioned combination of (TD1) with sequence (d)(a)(r), seems to be recommended (compare Table 12).

With unrestricted resources for computing time and memory, we would recommend applying trip distribution constraints with a short time-period for both synthetic data and real-world data. When computing resources are limited and quick decisions are required, we recommend introducing event sequences (r)(d)(a) or (d)(a)(r). Within the experiments on real-world data, partially-integer variable domains, along with constraints (TD1) and event sequence (d)(a)(r) performed particularly well.

Note that we have used a time limit of 24h in order to be capable of solving the models to optimality, allowing us to draw conclusions on their degree of realism and potential performance in practice. Solving models multiple times throughout the day (albeit over a smaller time horizon) in sufficiently short computing times may require the use of parallel computing, specialized solution methods (e.g., mathematical decomposition), or a combination of both.

6.2 Future work

Having explored the different modeling assumptions and techniques both from methodological and empirical standpoints, this work aimed at shedding light on the modeling jungle and guiding both practitioners and academics in future research on multi-period rebalancing optimization.

While concepts such as the sequences of events cannot be found in most of the related classical optimization problems, such as the Pickup-and-Delivery Problem, such characteristics are not exclusive to

BSSs. Indeed, any planning problem in which the interaction between customers and system operator is aggregated in discretized time-periods (e.g., car-sharing, multi-mode transportation planning with synchronization) may exhibit similar event sequences, which may be worth studying.

Certain future research directions may be particularly worthwhile. First, while we have identified the formulations that are likely to provide well-performing planning solutions, solving those models in real-time throughout the day may be challenging; therefore, decomposition algorithms may be employed to speed up the solution time. Second, the proposed models minimize total lost demand. Certain BSS operators consider target intervals, which may be interesting to consider within the objective function.

Finally, while we have intentionally focused on the deterministic planning problem which assumes as input a single set of customer demands, some of our conclusions may also hold for models that consider several demand scenarios simultaneously. Models explicitly considering the underlying uncertainty and demand probability distribution are worthwhile research directions, in particular for the model variants that have shown to be more realistic here.

Appendix

1 Other assumptions and objective functions

1.1 Initial location and inventory of vehicles

At the beginning of the planning horizon, vehicles are located at specific stations or a depot with a predefined inventory. Existing works assume that the initial locations and inventories of vehicles are always known and fixed [see, e.g., 8, 9, 14]. The total number of available bikes for rebalancing may be uniformly or arbitrarily assigned to each vehicle. In contrast, some studies, such as [10] and [31], consider a depot and use it as the initial location of the vehicles.

Intuitively, better solutions may be obtained if the model can explicitly decide on the initial location and inventory of the vehicles [see, e.g., 29].

We numerically explore the impact of these initial settings. Given that these settings rather concern the definition of the planning problem, but not the modeling assumptions, the surrounding discussions can be found in Appendix 6.1. Throughout all other experiments, we applied fixed vehicle location and inventory as it has been common in the literature.

Finally, to summarize the various modeling assumptions used for the existing multi-period models in the literature, Table 14 classifies the existing works by their various alternatives.

1.2 Objective functions

Decision-makers may have different objectives for their rebalancing strategies. The most common objective for dynamic rebalancing is to minimize lost demand, which is used within our experiments. The objective is part of the problem definition and not within the scope of our paper. Further, models with different objectives are difficult to compare directly. Therefore, we refrain from evaluating the impact of using different objectives in this paper.

We summarize the various objectives in Table 15, classifying the measurements into several main aspects and marking with '✓' when the objective function of the corresponding reference contains a particular aspect. We also emphasize the modeling techniques the analyzed papers used to respect the rebalancing problem, mainly including Linear Programming (LP), MILP, Mixed-integer Nonlinear Programming (MINLP), Constraint Programming (CP), Neural Networks (NN), and MDP. Some models may be non-linear since they have non-linear terms in the objective function or in some constraints.

Table 14: Modeling assumptions in MIP-based multi-period rebalancing models

References	Time Constraints		Initial Location and Load of Trucks		Trip Distribution		Sequences of Events	Variable Type
	Traveling	Handling	Fixed	Flexible	Proportionality	Poisson		
[8]	✓		✓				(r+a+d)	Station-based
[9]			✓		✓		(r+a+d)	O-D
[13, 30]	✓	✓	✓				(a)(r)(d)	Station-based
[10]	✓	✓	✓				(a+d)(r)	Station-based
[14]	✓		✓		✓		(a)(d+r)	Station-based
[29]				✓		✓	(r+a+d)	O-D
[36]	✓	✓	✓		✓		(r+a+d)	O-D
[31]	✓	✓	✓				(r)(a+d)	Station-based
[27]			✓		✓		(r+a+d)	O-D

Concerning the criterion used within the objective function, distance-based metrics are associated with the traveling distance of vehicles, mainly including traveling cost, traveling time, and fuel consumption. Loading-based metrics are associated with the number of handling (loading/unloading) operations. Researchers normally consider handling costs or time in the objective functions, which reflects the required workload of such operations. Two metrics aim at representing the dissatisfaction of customers: one minimizes the deviations from a target value, while another minimizes the lost demand (or, equivalently, maximizes the successful trips). Besides these popular metrics, some other factors have been considered, such as the cost of holding bikes by rebalancing vehicles, parking costs, CO_2 emissions, costs of using trucks (related to the number of trucks employed), and the number of visits of full vehicle loads. For example, [53] consider the number of bikes held by the trucks during each rebalancing movement and include the total holding cost in the objective function. [45] add to the objective the usage cost of employing each truck for rebalancing. [24] add to the objective a parking time for each station visit using the instances from an operator in Norway. [55] consider only full rebalancing vehicle loads among stations and maximize the total number of full vehicle loads picked up and delivered to the stations, which is indicated as the number of visits in Table 15.

Given that SBRP mainly focus on night-time operation scenarios, where the dynamic demand is less important, their objectives mainly aim at minimizing the costs of the rebalancing operations based on traveling distance, the number of loading/unloading operations, and deviations from predefined target numbers of bikes (see Table 15. Note that some works exclude the deviation from target values or satisfied demand from the objectives, but implement particular constraints to guarantee the satisfaction of user demand to some degree [49, 51, 60, et al.].

In DBRP, rebalancing operations are performed multiple times during the day and real-time trip flow is considered when rebalancing takes place. Due to the more complex nature of DBRP, their objectives tend to include more aspects considered by operators, especially concerning the demand unsatisfaction. Generally, the existing objective functions quantify the distance-based metrics and customers' satisfaction, including the traveling cost, handling cost, and lost demand. Some focus on maximizing the profits of successful trips [e.g. 9]. [36] consider jointly traveling cost and handling operations, which adds immediate value to the rebalancing. Among the studies considering more than one aspect in their objective functions, [10] and [36] attribute a weight to each of them.

Table 15: Objectives and modeling technique of related literature

		Reference	Objective fuctions: Distance-based metrics[1]	Loading-based metrics[2]	Number of Bikes in truck	Park-ing	CO_2	Target Value Dev[3]	Lost demand/ Satisfied demand	Number of trucks	Number of visits	Modeling Technique
Static		[45]	✓					✓				MILP
		[46]	✓									MILP
		[47]	✓									MILP
		[48]										MILP
		[49]	✓									MILP
		[50]	✓									NN
		[51]	✓									MILP
		[52]	✓	✓				✓				CP
		[53]	✓		✓							MILP
		[54]	✓									MILP
		[24]	✓	✓		✓		✓				MILP
		[39]	✓						✓			MILP
		[55]									✓	MILP
		[56]	✓									MINLP
		[57]	✓					✓				MILP
		[58, 59]	✓	✓				✓				Combinatorial
		[23]	✓						✓			MILP
		[60]	✓	✓								MILP
		[61]	✓	✓				✓				MINLP
		[62]	✓	✓				✓				MILP
		[63]	✓							✓		MILP
Dynamic	Multi-period	[1]							✓			MDP
		[8]							✓			MILP
		[9]	✓						✓			MILP
		[10]	✓	✓				✓	✓			MINLP
		[4]							✓			MDP
		[14]							✓			MILP
		[29]			✓				✓			LP
		[33]	✓						✓			LP
		[31]	✓						✓	✓		MILP
		[27]	✓						✓			MILP
	Rolling Period	[13, 30]							✓			MILP
		[64]						✓				MILP
		[65]	✓	✓					✓	✓		MILP
		[36]	✓						✓			MILP
		[66]					✓		✓			Combinatorial
		[67]						✓				MINLP
		[32]	✓						✓			MINLP

[1] These factors are based on the traveling distance, such as traveling cost, traveling time, and fuel consumption.
[2] These factors are based on the number of loading/unloading operations, such as handling cost and handling time.
[3] Deviations from a target value.

2 Proportionality constraints

In the case where O-D variables $x^{t_1,t_2}_{s_1,s_2}$ are used to represent successful trips, Constraints (TD5) are automatically guaranteed. The proportional distribution can then be written as

$$x^{t_1,t_2}_{s_1,s_2} \leq ab^{t_1}_{s_1} \frac{F^{t_1,t_2}_{s_1,s_2}}{f^{+,t_1}_{s_1}} \qquad \forall\ s_1, s_2 \in S, t_1, t_2 \in T. \tag{TD7}$$

Constraints (TD7) imply that rentals from a station have to respect the transition probability when rental demand exceeds the number of available bikes at that station. [9] use such constraints with $ab^{t_1}_{s_1} = d^{t_1}_{s_1}$.

Note that summing (TD7) over t_1 and s_1, the left-hand side becomes $x^{-,t_2}_{s_2}$, which results in (TD3). Thus, station-based trip variables with proportional distribution and O-D variables with proportional distribution essentially represent the same type of trip distribution within the model.

For O-D variables in the Practical toy example of Section 3.2.2, the constraints enforcing the flow to be no more than the demand of each route and the returns to be less than the available docks can be combined as $x^{t_1,t_2}_{s_1,s_2} \leq \min\{F^{t_1,t_2}_{s_1,s_2}, ad^{t_2}_{s_2}\}$. To facilitate the comparison of the solutions, we present the solution $[x^{t_1,t_2}_{s_1,s_3}, x^{t_1,t_2}_{s_1,s_4}]$ of the O-D variables model in the equivalent format of station-based trip variables, e.g., $x^{+,t_1}_{s_1} = x^{t_1,t_2}_{s_1,s_3} + x^{t_1,t_2}_{s_1,s_4}$. The results are shown in Table 16.

Table 16: Trip distribution for 3 different demand scenarios under different trip distribution constraints (O-D variables)

	Constraints	I-I	I-S	S-I
Ideal solution		[*, *, 4, 2]	[4, 3, *, *]	[*, *, 4, 2]
O-D Variables	Without (TD7)	[4, 2, $x^{-,t_2}_{s_3} + x^{-,t_2}_{s_4} = 6$]	[4, 3, $x^{-,t_2}_{s_3} + x^{-,t_2}_{s_4} = 7$] $x^{t_1,t_2}_{s_2,s_3} = 1, x^{t_1,t_2}_{s_2,s_4} = 2$	[$x^{+,t_1}_{s_1} + x^{+,t_1}_{s_2} = 6$, 4, 2]
	With (TD7)	[$x^{+,t_1}_{s_1} + x^{+,t_1}_{s_2} = 5$, 3, 2] $x^{t_1,t_2}_{s_1,s_3} = 2, x^{t_1,t_2}_{s_2,s_3} = 1$	[4, 3, 3, 4] $x^{t_1,t_2}_{s_1,s_3} = 2, x^{t_1,t_2}_{s_1,s_4} = 2$	[$x^{+,t_1}_{s_1} + x^{+,t_1}_{s_2} = 6$, 4, 2]

For O-D variables, constraints (TD7) mainly work for the case, where the rental demands cannot be satisfied. [8] applied station-based variable without any trip distribution constraints. [9] use (TD7). [14] use constraints that are similar to Constraints (TD1).

We give another practical toy example here. The 4 different configurations are shown in Figure 5.

Practical toy example with 3 stations

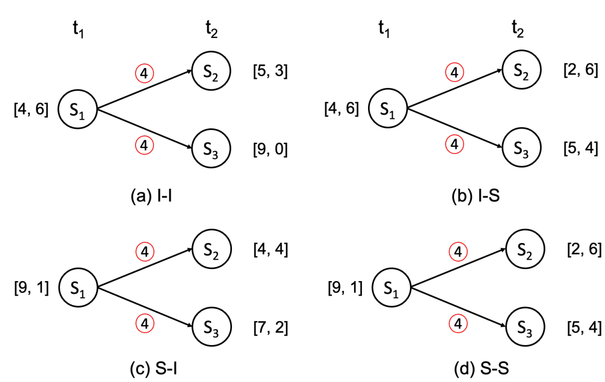

Figure 5: Trip flow example with 3 stations

Table 17: Solutions under different circumstances

$[x_{s_1}^{+,t_1}, x_{s_2}^{-,t_2}, x_{s_3}^{-,t_2}]$	Constraints	I-I	I-S	S-I
	Ideal solution	[3, 3, 0],[2, 2, 0]	[4, 2, 2]	[6, 4, 2]
Station-based variable	without TD	[4, 3, 0]	[4, 4, 4]	[8, 4, 2]
	(TD1)	[4, 2, 0]	**[4, 2, 2]**	[8, 4, 2]
	(TD2)	**[3, 3, 0]**	[4, 4, 4]	**[6, 4, 2]**
	(TD6)	**[3, 3, 0]**	[4, $x_{s_2}^{-,t_2} + x_{s_3}^{-,t_2} = 4$]	**[6, 4, 2]**
	(TD1)+(TD2)	[0, 0, 0]	**[4, 2, 2]**	[4, 2, 2]
	(TD1)+(TD6)	[0, 0, 0]	**[4, 2, 2]**	[4, 2, 2]
	(TD2)+(TD6)	**[3, 3, 0]**	[4, $x_{s_2}^{-,t_2} + x_{s_3}^{-,t_2} = 4$]	**[6, 4, 2]**
	(TD3)	[4, 2, 0]	**[4, 2, 2]**	[8, 4, 2]
	(TD4)	**[3, 3, 0]**	[4, 4, 4]	**[6, 4, 2]**
	(TD3)+(TD4)	[3, 2, 0]	**[4, 2, 2]**	**[6, 4, 2]**
	(TD3)+(TD6)	**[2, 2, 0]**	**[4, 2, 2]**	**[6, 4, 2]**
	(TD1)+(TD4)	[3, 1.5, 0]	**[4, 2, 2]**	[6, 3, 2]
	(TD2)+(TD3)	**[2, 2, 0]**	**[4, 2, 2]**	**[6, 4, 2]**
O-D variable	without (TD7)	**[3, 3, 0]**	[4, $x_{s_2}^{-,t_2} + x_{s_3}^{-,t_2} = 4$]	**[6, 4, 2]**
	(TD7)	**[2, 2, 0]**	**[4, 2, 2]**	**[6, 4, 2]**

As we can see from Table 17, constraints (TD2) add a strict limitation to the trip flow when there is no dock available in one of the stations. If one of the return variables can be realized is zero (the station is full), then it may cause all the variables equal to zero based on (TD1) and (TD2). In all 4 circumstances, the solutions under constraints (TD6), (TD2)+(TD6) with station-based variables, and no proportional constraints with O-D variables can be the same. Furthermore, the solution under constraints (TD3)+(TD6) is equivalent to the solution under constraints (TD7). Generally, (TD2)+(TD3) and (TD3)+(TD6) perform well in this case. (TD6), (TD4), and (TD2) also give a reasonable solution.

3 Formulations for event sequences

The rest of the event sequences are demonstrated here.

- **(a)(d)(r):** Here, rebalancing operations occur at the end of each time-period, which is enforced by Constraints (18)–(21).

$$\sum_v r_{s,v}^{+,t} \le d_s^t + x_s^{-,t} - x_s^{+,t} \qquad \forall\ s \in S,\ t \in T \tag{18}$$

$$\sum_v r_{s,v}^{-,t} \le C_s - d_s^t - x_s^{-,t} + x_s^{+,t} \qquad \forall\ s \in S,\ t \in T \tag{19}$$

$$x_s^{+,t} \le d_s^t + x_s^{-,t} \qquad \forall\ s \in S,\ t \in T \tag{20}$$

$$x_s^{-,t} \le C_s - d_s^t \qquad \forall\ s \in S,\ t \in T \tag{21}$$

- **(d)(r)(a):** Rebalancing operations occur between rentals and returns, which is enforced by Constraints (22) and (23). Bike rentals occur at the beginning of the period and are only restricted by the current capacity of the station, as indicated by Constraints (24). Bike returns occur at the end and are limited by $x_s^{-,t} \le C_s - d_s^t + \sum_v r_{s,v}^{+,t} - \sum_v r_{s,v}^{-,t} + x_s^{+,t}$, which can be achieved by replacing d_s^{t+1} in Constraints (6) ($d_s^{t+1} \le C_s$) with the right-hand side of Constraints (3).

$$\sum_v r_{s,v}^{+,t} \le d_s^t - x_s^{+,t} \qquad \forall\ s \in S,\ t \in T \tag{22}$$

$$\sum_v r_{s,v}^{-,t} \le C_s - d_s^t + x_s^{+,t} \qquad \forall\ s \in S,\ t \in T \tag{23}$$

$$x_s^{+,t} \le d_s^t \qquad \forall\ s \in S,\ t \in T \tag{24}$$

For the class of event sequences reported in the second row of Table 1, the corresponding modifications are as follows:

- **(d)(a)(r):** We first use Constraints (24) since rentals occur first. Then, to implement correct returns, we add Constraint (25). Finally, given that the vehicles are assumed to rebalance after bike rentals and returns, we also use vehicle constraints (18) and (19).

$$x_s^{-,t} \leq C_s - d_s^t + x_s^{+,t} \qquad \forall\ s \in S,\ t \in T \tag{25}$$

- **(a)(r)(d):** Returns occur at the beginning of the period and consider the current inventory, as ensured by Constraints (21). We also add Constraints (26) and (27), since vehicles rebalance bikes after returns. As customers rent bikes at the end of each time-period, we have to enforce $x_s^{+,t} \leq d_s^t - \sum_v r_{s,v}^{+,t} + \sum_v r_{s,v}^{-,t} + x_s^{-,t}$, which is explicitly satisfied when replacing d_s^{t+1} in Constraints (6) ($0 \leq d_s^{t+1}$) by the right hand side of Constraints (3).

$$\sum_v r_{s,v}^{+,t} \leq d_s^t + x_s^{-,t} \qquad \forall\ s \in S,\ t \in T \tag{26}$$

$$\sum_v r_{s,v}^{-,t} \leq C_s - d_s^t - x_s^{-,t} \qquad \forall\ s \in S,\ t \in T \tag{27}$$

We now consider event sequences in which rentals and returns are assumed to happen simultaneously (see the third row in Table 1):

- **(a+d)(r):** The constraints for rebalancing are the same as for **(a)(d)(r)** and **(d)(a)(r)**, i.e., Constraints (18) and (19). The restrictions for rentals and returns are enforced by using Constraints (20) and (25).

Finally, two more classes of event sequences allow for rebalancing to occur simultaneously with either rentals or returns (see the fourth and fifth rows respectively in Table 1).

- **(a)(d+r):** Customers return bikes first, requiring the use of Constraints (21). Rebalancing then simultaneously occurs with rentals, which can be implemented using Constraints (18) and (19).
- **(d+r)(a):** Here, employing Constraints (22) and (23) is sufficient.
- **(d)(r+a):** Customers rent bikes first, requiring Constraints (24). Rebalancing then simultaneously occurs with returns, requiring Constraints (18) and (19).
- **(r+a)(d):** It suffices to add Constraints (26) and (27).

4 Generation of problem instances

In this appendix, we provide more details on the generator for the synthetic problem instances.

4.1 Station network and operating settings

We first generate the station network. The parameters are defined in Table 18. We consider a rectangular study area defined by the latitude and longitude values for each of its 4 vertices. This area is divided into a total of *num_grid* grids, each of which can be assigned to at most one station. In our studies, we use a rectangular study area with 150 × 150 grids (i.e., *num_grid* = 22500) with latitude values from 45.4 to 45.65 and longitude values from -73.71 to -73.49 (approximating the Montreal island area). The total number of stations, the number of city centers, and the total capacity of the station network can be set and changed according to what kind of instances we focus on.

We assume that there are either one or two city centers in the study area, including stations that have high return demands during morning peak hours and high rental demands during afternoon peak hours. If there is only one city center, its central grid is randomly selected within a square spanning grid 53 to 98 on both the x- and the y-axis of the study area. If there are two city centers, one central grid is selected randomly within the square spanning grids from 30 to 75, and the other one within the square spanning grids from 75 to 120 on both the x- and the y-axis of the study area. Each city center is then defined as an area of $ran\_cc \times ran\_cc$ grids around its central grid.

Table 18: The input and output of station generation

Input	Study area: minimum longitude and latitude, maximum longitude and latitude
	num_grid: the number of grids *num_station*: total number of stations *num_cc*: the number of city centers *total_c*: the total capacity of all the stations *ran_cc*: the range of city centers (in number of grids) *per_cc*: the proportion of the total network capacity located within city centers *cap_cs*: capacity of a city center station *cap_os*: capacity of a regular station
Output	Locations of stations Distances between station pairs Number of city center stations Number of regular stations

Each station network has a total of *num_station* stations (set either to 30 or 60). Regular stations are assumed to have a capacity of $cap\_os = 20$ docks. Given that city center stations typically have a much larger capacity, we here assume a capacity of $cap\_cs = 40$ docks for each city center station. We consider that the total capacity of all the stations (i.e., the number of docks in the entire network) is roughly proportional to the total number of stations *num_station*, as observed within the network of BIXI Montreal which had a total capacity of 14,078 with 617 stations in 2019. Therefore, we set the total capacity *total_c* to 1,369 for 60 station networks and 685 for 30 station networks. Note that *total_c* will be used only to compute the number of regular and city center stations. The number of city center stations is set to $\lfloor total\_c \times per\_cc / cap\_cs \rceil$ (where *per_cc* is defined in Table 5 for each of the ground truths). The remaining stations are assumed to be regular (i.e., out of the city center) stations. We then randomly assign stations to the grids as follows:

- **City center stations:** We randomly select grids inside the city center area as locations for city center stations. We show an example with 60 stations and two city centers in Figure 6. The dotted boxes are the range of city centers and the blue dots are the city center stations.
- **Regular stations:** The remaining stations will be randomly assigned to the grids outside the city center areas, indicated as green dots in Figure 6.

Once the locations of stations are fixed, we compute the distance between each station pair.

For the operating settings, we define the parameters in Table 19.

Table 19: The settings of BSSs and rebalancing fleet

n_bikes	the total number of bikes in the stations
n_trucks	the number of trucks available for rebalancing
cap_truck	the capacity of each truck
n_bikeT	the total available number of bikes for rebalancing trucks at the beginning

Finally, the total number of available bikes in the system, *n_bikes*, is 608, which is proportional to those observed at BIXI Montreal in 2019. We assume that 4 trucks (*n_trucks*) are available to rebalance the bikes. The capacity of each truck, *cap_truck*, is set to 40 and the number of available bikes for trucks to employ, *n_bikeT*, is set to 80.

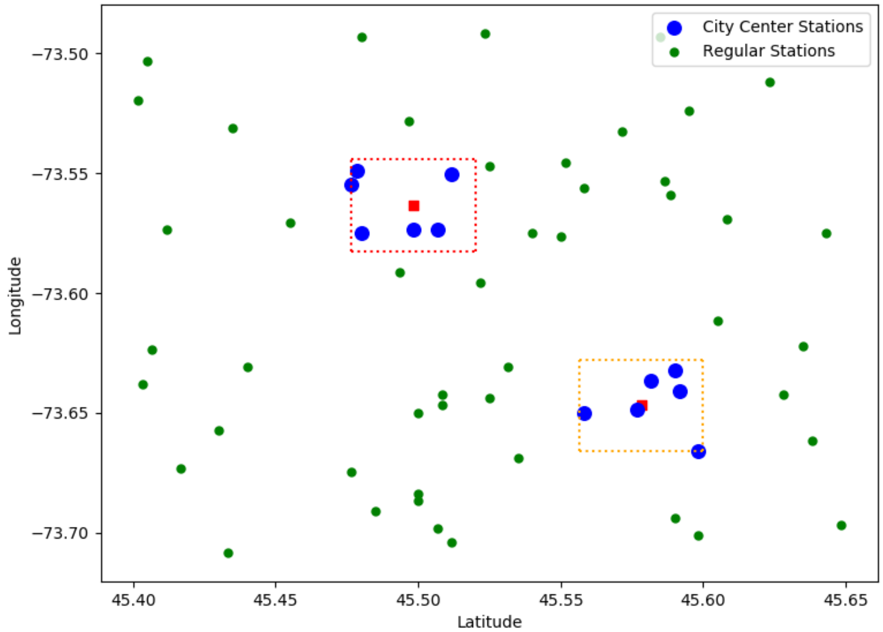

Figure 6: Visualization of station network used by GT2 with 2 city centers, indicating regular stations (green dots), city center stations (blue dots), and the city center central grid (red rectangle). Note that each grid has a rectangular form.

4.2 Bike trips

Based on the generated station information, we generate the trip data. Analyzing the trip data from BIXI, we found that the demand has a similar pattern on weekdays with a morning peak and an afternoon peak that are mainly caused by work-related trips. At weekends, there is less regularity and trips seem more random. We therefore focus on weekdays only. To provide adequate but diverse problem instances and to fully explore the impacts, we generate trips based on real-world data with adaptable parameters, instead of directly applying real-world data.

Parameters to define trip demands. The parameters used for the trip generation are defined in Table 20. Each trip contains an origin station, a destination station, a departure time, and an arrival time. We set an average total number of trips avg_trips per weekday, which can be estimated from historical trip data in real BSSs. Note that $per\_io + per\_oo + per\_rd + per\_rn = 100$. After having the fixed number of trips of different types, we generate the trip data according to their characteristics. We assume that the departure time of each trip type follows a particular distribution (Dis), specifying the probability that a trip starts at a specific time. This allows us to model demand changes throughout the day while preserving uncertainty.

Table 20: The input and output of trips generation

Input	avg_trip: The average number of trips per weekday per_oi: The percentage of OI trips per_oo: The percentage of OO trips per_rd: The percentage of RD trips per_rn: The percentage of RN trips per_w: The percentage of work-related trips (OI and OO) expected to happen Dis: The set of distributions for different types of trips dur_trip: The interval for the length of trips
Output	All trips with origin stations, destination stations, departure time, and arrival time

The rules for trip generation are as follows:

- RD **and** RN **trips** We choose the origin station and destination station randomly among all stations for each random trip. The departure time is sampled from the corresponding distribution (one distribution for RD trips and one for RN trips). The duration of the trip is selected from the interval dur_trip at uniformly random, allowing us to also compute the arrival time for this trip. RD and RN trips are assumed to be one-way without any corresponding returns, as opposed to work-related trips. We repeat this process to obtain all required trips for each day. The total number of RD and RN trips will be $avg\_trips * per\_rd$ and $avg\_trips * per\_rn$ for each weekday.
- OI **trips and** OO **trips** Work-related trips normally have stable O-D pairs that include one trip from home to work and another one back to home with the same origin/destination stations. For each O-D pair, we first generate one trip from home to work. The stations are selected randomly according to the type of trips. For example, the origin station of an OI trip is selected randomly among the stations outside city centers, and the destination station is chosen from city center stations. Then, a corresponding return trip is generated with the origin and destination stations reversed. The process is the same for OO trips, except that both the origin and destination stations need to be selected among the stations outside city centers. The departure time obeys the assumed distribution. Differently from random trips, work-related trips happen during morning peak hours and afternoon peak hours. Thus, we use two different distributions for OI trips and two distributions for OO trips to imitate the two peaks. Using the departure time under the particular distribution and duration dur_trip, the arrival time is computed.

Random trips RD and RN may vary a lot from one day to another, so we generate them from scratch for each weekday. However, work-related trips have distinctive characteristics. They do not vary too much since users tend to commute between the same O-D pairs. Nevertheless, the demand for each weekday may change slightly because users may choose alternative means of transportation or not go to work for some personal reason on certain days. Thus, we consider an additional processing step for work-related trips. We generate a total of $avg\_trips * per\_oi$ OI and $avg\_trips * per\_oo$ OO trips that form a work-related trip set and we fix this set for all weekdays of a given problem instance. We then consider a probability per_w that a person is actually taking the bike for a route. For each day, the final demand for work-related trips is based on this fixed set. We then uniformly sample a random value between 0 to 1 for each work-related trip in the set. If the value is smaller than or equal to per_w, we select the corresponding trip with its rental and return demand. The actual number of work-related trips OI and OO at each weekday will slightly vary but will average to about $avg\_trips * per\_oi * per\_w$ and $avg\_trip * per\_oo * per\_w$, respectively.

Parameter values to generate trips. Since BIXI has a total number of around 33,300 trips per weekday on a 617 stations network, we set avg_trip to 3,240 for our 60 station network. The probability per_w for work-related trips is set to 0.85, which means that a user has an 85% chance to choose the bike, and its demand is generated. We assume that the duration of each trip is within $[5, 30]$ minutes (dur_trip).

We now describe the settings of the distributions in Dis. We illustrate the weekday average demand of one week at BIXI Montreal from July 2019 in Figure 7(a). An example of generated synthetic data is demonstrated in Figure 7(b)–(f), averaging over 500 days for 24 hours discretized into 48 time-periods. Since our work mainly focuses on rebalancing optimization, we use Beta distributions and linear transformations to fit our demand curve to the one of BIXI. We apply two Beta distributions ($\alpha = 3, \beta = 8$) for the departure time of OI trips in morning peak hours and afternoon peak hours. The random value x generated by the Beta distribution is shifted into departure time using a linear transformation $ax + b$. For example, we set $a = 530$ and $b = 340$ for the OI trips during the morning hours to shift random numbers into the interval of [340, 870] minutes. After transformation, we obtain Figure 7(c) with two peaks representing the morning and evening rush hours. Similarly, two Beta distributions ($\alpha = 3, \beta = 7$) with $a = 550$ and $b = 900$ for OO trips are employed, as shown in Figure 7(d).

For RD trips happening around 10 a.m. - 9 p.m., we use a Beta distribution ($\alpha = 3, \beta = 7$) with $a = 900$ and $b = 560$, as illustrated in Figure 7(e). A Beta distributions ($\alpha = 6, \beta = 8$) with $a = 1200$ and $b = 750$ is shifted to represent RN trips, mainly during 4 p.m. - 5 a.m., as depicted in Figure 7(f). Summing the demand of the four trip types, Figure 7(b) illustrates the total average demand over 500 days.

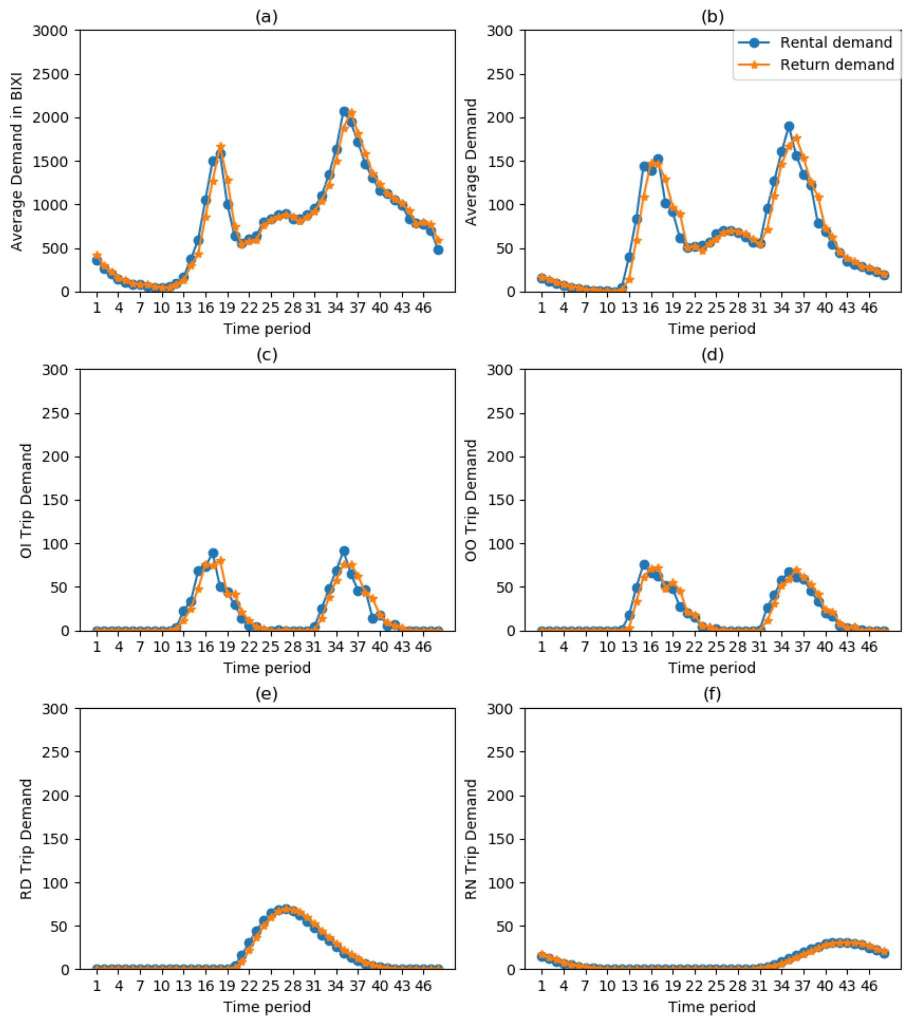

Figure 7: Demand Information for 500 days in 24 hours (48 time-periods)

4.3 Ground truths for experiments

To test the models under different station environments and trip patterns, we generate 3 ground truths based on the above-explained settings of parameters. Some of the parameters have the same values in the three ground truths, namely:

- num_grid: 150x150
- $num\_station$: 60
- n_trucks: 4
- cap_truck :40
- n_bikeT : 80
- avg_trip: 3630
- per_w: 85%.

The parameters specified for each of the ground truths are defined in Table 5 in Section 5.1.

5 Pseudo-code for the simulator

The pseudo-code for our simulator is given in Algorithm 1. The initial inventory of each station d_s^0 and each truck d_v^0 are given as inputs, along with the capacity for each station C_s and each truck $\hat{C}_v$. Rebalancing strategies $r\_pick(s,t,v)$ and $r\_drop(s,t,v)$ are obtained from the optimization model. Trip sequence N is composed of individual trips $\{[t_d(n), s_d(n), t_a(n), s_a(n)], n \in N\}$, including rental and return demands with $t_d(n)$ ascending. For a trip n, $t_d(n)$ is the time of departure, $s_d(n)$ is the departure station, $t_a(n)$ is the arriving time, and $s_a(n)$ is the arrival station. A waiting set of M events $W = \{[w_t(m), w_s(m), w_i(m), w_v(m)], m \in M\}$ stores the upcoming demands and rebalancing operations. The elements of W change with real-time system status. Each element $w \in W$ has an indicator $w_i(m)$ that either represents a rental demand ($w_i(m) = d$), a return demand ($w_i(m) = a$), a pick-up operation ($w_i(m) = p$), or a drop-off operation ($w_i(m) = f$). Moreover, $w_t(m)$ is the departure time of the event, $w_s(m)$ is the station, and $w_v(m)$ is the truck. Note that the value of $w_v(m)$ has no impact on demand events and is set to 0. The waiting set will be sorted in non-decreasing order of $w_t(m)$ and its first element is the event that will be processed next. Parameter $D_{s,s'}$ represents the distance between two stations s and s'. The transit time (in minutes) between two stations during time-period t is given by $R_{s,s'}^t$ and the average operation time for picking up or dropping off one bike is given by op.

We denote the inventory of the station s as $Avails\_bike(s)$ and the inventory of the vehicle v as $Availv\_bike(v)$. The lost demand during a period t for a station s will be counted in $Lost\_rental(t,s)$ and $Lost\_return(t,s)$ respectively. We first initialize W with all rental demands from N with $w_i = d$ and $w_v = 0$.

The corresponding returns of successful rentals and rebalancing events are created and added to W in simulated real-time. For each truck and time-period, we create $r\_pick(s,t,v)/r\_drop(s,t,v)$ consecutive events with ($w_i = p$ or $w_i = f$ respectively). Rebalancing starts as soon as the truck arrives at the station, but not before the first minute associated with time-period t. A truck leaves for the next station as soon as it finishes the rebalancing operations at the current station. If the truck reaches the next station before the end of the current time-period, it waits until the beginning of the next time-period before starting the rebalancing operations. Since rebalancing and relocation may be scheduled continuously one after another, some rebalancing operations may be delayed due to the previous operations.

As in reality, the simulator processes the events in W in chronological order. When a rental demand occurs and the station holds at least one available bike, we update the station inventory and add the corresponding return demand to the waiting set W. Otherwise, the customer is assumed to leave the system and a lost rental demand is counted. When a return demand occurs but the station has no available docks, we assume that the customer returns the bike at the nearest station with available docks. However, a lost return demand will be counted. For pick-up/drop-off rebalancing operation events, we verify whether sufficient bikes/docks at the station and space/bikes within the truck are available. Rebalancing is carried out as close as possible to the originally planned operations. The inventories of the station and the truck are updated accordingly. After partially/fully successive rebalancing, the truck departs for the next station.

6 Complimentary experiments

6.1 Initial settings for vehicles

In our basic model, the initial location and inventory of each vehicle are fixed. The fixed initial locations are at stations 1, 16, 31, and 46. Each truck has the same amount of bikes, i.e., 20 bikes. However, the operator may have the possibility and desire to specify an initial location and inventory for the trucks. To this end, Constraints (28) and (29) are created. Constraint (28) implies that the

Algorithm 1: Simulator for the rebalancing strategy $r\_pick(s,t)$ and $r\_drop(s,t)$

Input : $\{[t_d(n), s_d(n), t_a(n), s_a(n)], n \in N\}$, $D_{s,s'}$, $r\_pick(s,t)$, $r\_drop(s,t)$, d_s^0, d_v^0, C_s, $\hat{C}_v$, $R^t_{s,s'}$, and op. T is the planning horizon.

Initialization: $Lost\_rental(t,s) = 0$; $Lost\_return(t,s) = 0$; $Avails\_bike(s) = d_s^0$; $Availv\_bike(v) = d_v^0$; $W = \{[w_t(m), w_s(m), w_i(m), w_v(m)], m \in M\}$ with all the rental demands from N and the operation events of the first time-period sorted; $time = w_t(1)$; $s = w_s(1)$; $indicator = w_i(1)$; and $v = w_v(1)$.

$n = 1$;
while $time \le T$ *and* $W \neq \emptyset$ **do**
  Find corresponding time-period t based on $time$;
  $sign = 0$;
  if $indicator = d$ **then**
    if $Avails\_bike(s) > 0$ **then**
      $Avails\_bike(s) = Avails\_bike(s) - 1$;
      $W = W \cup \{[t_a(n), s_a(n), a, 0]\}$;
    else
      $Lost\_rental(t,s) = Lost\_rental(t,s) + 1$;
    $n = n + 1$;
  else if $indicator = a$ **then**
    if $C_s - Avails\_bike(s) > 0$ **then**
      $Avails\_bike(s) = Avails\_bike(s) + 1$;
    else
      $Lost\_return(t,s) = Lost\_return(t,s) + 1$;
      Find s' closest to s with available docks based on $D_{s,s'}$;
      $Avails\_bike(s') = Avails\_bike(s') + 1$;
  else if $indicator = p$ **then**
    if $Avails\_bike(s) > 0$ *and* $Availv\_bike(v) < \hat{C}_v$ **then**
      $Avails\_bike(s) = Avails\_bike(s) - 1$;
      $Availv\_bike(v) = Availv\_bike(v) + 1$;
      if *All the rebalancing operations are done for* s **then**
        $sign = 1$;
    else
      Remove the elements in W whose $w_s = s, w_i = p, w_v = v$;
      $sign = 1$;
  else
    if $Availv\_bike(v) > 0$ *and* $Avails\_bike(s) < C_s$ **then**
      $Avails\_bike(s) = Avails\_bike(s) + 1$;
      $Availv\_bike(v) = Availv\_bike(v) - 1$;
      if *All the rebalancing operations are done for* s **then**
        $sign = 1$;
    else
      Remove the elements in W whose $w_s = s, w_i = f, w_v = v$;
      $sign = 1$;
  if $sign = 1$ **then**
    Create W' of v for $t+1$ based on $R^t_{s,s'}$, and op;
    $W = W \cup W'$;
  $W = W \backslash \{[time, s, indicator, v]\}$;
  $time, s, indicator, v = w_t(m), w_s(m), w_i(m), w_v(m)$ where $w_t(m)$ is the minimum in W;
end

Output : $Lost\_rental(t,s)$ and $Lost\_return(t,s)$

number of vehicles located at specific stations at the first time-period is equal to the number of vehicles num_v in the system, which, along with the Constraints (4), means that the trucks can be assigned to any station at the beginning of rebalancing. Constraint (29) indicates that the total number of bikes in all vehicles equals the total number of bikes av_bike initially available in vehicles. Here, av_bike is set to 80, while the inventory at each vehicle is optimized.

$$\sum_{s,v} z^1_{s,v} = num\_v \tag{28}$$

$$\sum_{v} \hat{d}^1_v = av\_bike \tag{29}$$

Note that, if we need to consider a central depot in our system, this depot can be represented as an additional station with a particular capacity in our optimization model.

For experiments, we consider 30-minute time-periods in the station-based model without trip distribution constraints. Table 21 summarizes the results for 3 Ground truths. Compared to the initial inventory, the initial location has obvious impacts on lost demand for dynamic rebalancing. The case with fixed initial inventory and flexible location has the best performance for lost rentals. The results highlight that the initial location of the trucks is important, assuming that each truck holds sufficient bikes. For GT3 with too many work-related trips, the models are hard to solve to optimality. The trip demand is highly concentrated during the peak hours, which makes MIP gaps hard to reach 0.01%. Under these MIP gaps, the system can still benefit from the flexibility of initial locations.

Table 21: Station-based Model with different initial vehicle settings (60 stations, 80 available bikes for 4 trucks, 30 mins)

	Initial Location	Initial Inventory	O.F. Value	Time (mins)	MIP Gap (%)	Lost Demand (%)	
						Rental	Return
GT1	Fixed	Fixed	0.8	<1	0.00	8.78	7.99
	Fixed	Flexible	0.8	<1	0.00	8.39	8.07
	Flexible	Fixed	0.1	<1	0.00	8.26	5.89
	Flexible	Flexible	0.1	<1	0.00	8.30	5.60
GT2	Fixed	Fixed	0.5	<1	0.00	9.46	1.78
	Fixed	Flexible	0.5	<1	0.00	9.59	2.37
	Flexible	Fixed	0.2	<1	0.00	8.50	1.93
	Flexible	Flexible	0.1	<1	0.00	8.92	1.58
GT3	Fixed	Fixed	179.1	1440	2.46	20.33	21.67
	Fixed	Flexible	177.1	1440	2.33	19.89	20.86
	Flexible	Fixed	162.1	1440	2.52	17.46	16.66
	Flexible	Flexible	161.3	1440	2.37	17.78	17.19

When the BSSs network is small, a fixed initial setting is easier for the operators with an acceptable performance since no adjustment is needed for trucks before the beginning of dynamic rebalancing. However, if the station network is complex, the flexible initial setting may be beneficial to obtain a better performance of lost demand.

A similar conclusion is observed in Table 22 for the O-D variable model. The flexible initial location and fixed inventory give the best performance for GT1 and GT3. For GT2, the flexible location and flexible inventory obtains the best results. Since there are more city center stations in GT2 and the network is more complex, the flexibility of the rebalancing fleet has more advantages over the fixed one. In general, O-D variables seem to work better than the station-based trip variables for GT1 and GT3 with only one city center.

6.2 Initial station inventory and trip variable types

The static rebalancing (Baseline 2) model is presented as (30)–(34).

$$\min \quad \sum_{t,s}(f_s^{+,t} - x_s^{+,t}) + \sum_{t,s}(f_s^{-,t} - x_s^{-,t}) \tag{30}$$

$$d_s^{t+1} = d_s^t - x_s^{+,t} + x_s^{-,t} \qquad \forall\, s \in S,\ t \in T \tag{31}$$

$$\sum_s d_s^1 = n \tag{32}$$

$$0 \le d_s^t \le C_s \qquad \forall\, s \in S,\ t \in T \tag{33}$$

$$0 \le x_s^{+,t} \le f_s^{+,t}, 0 \le x_s^{-,t} \le f_s^{-,t}, \qquad \forall\, s \in S,\ t \in T \tag{34}$$

where n is the total number of bikes in the system.

The results for the O-D model with baseline 1 and baseline 2, as well as for GT3 are illustrated in Table 23.

Table 22: O-D model with different initial settings (60 stations, 80 available bikes for 4 trucks, 30 mins)

	Initial Location	Initial Inventory	O.F. Value	Time (mins)	MIP Gap (%)	Lost Demand (%)	
						Rental	Return
GT1	Fixed	Fixed	52.8	< 1	0.00	7.98	5.97
	Fixed	Flexible	52.8	< 1	0.00	8.31	8.40
	Flexible	Fixed	52.4	< 1	0.00	7.73	4.76
	Flexible	Flexible	52.5	< 1	0.00	8.38	7.68
GT2	Fixed	Fixed	51.5	288	0.00	9.95	2.27
	Fixed	Flexible	51.6	289	0.00	9.82	2.35
	Flexible	Fixed	50.9	< 1	0.00	9.36	2.07
	Flexible	Flexible	50.8	< 1	0.00	9.33	2.02
GT3	Fixed	Fixed	227.2	1440	2.44	19.79	21.33
	Fixed	Flexible	223.2	1440	2.53	19.92	21.19
	Flexible	Fixed	207.5	1440	2.74	16.00	18.72
	Flexible	Flexible	208.6	1440	2.64	17.08	18.32

Table 23: O-D model with baseline 1 and baseline 2 and station-based model for GT3 (60 stations, 4 trucks, 30 mins)

	Baselines, Configuration, Trip Modeling	O.F. Value	Opt. Time (mins)	MIP Gap (%)	Lost Demand (%)	
					Rental	Return
GT1	Baseline 2 dyn.rebal. O-D	52.8	< 1	0.00	7.98	5.97
GT2	Baseline 2 dyn.rebal. O-D	51.5	288	0.00	9.95	2.27
GT3	Baseline 1 without rebal.	-	-	-	33.26	33.83
	Baseline 2 without rebal. (static)	-	-	-	28.42	33.02
	Baseline 1 dyn.rebal. station-based	232.0	1440	2.60	23.24	18.24
	Baseline 2 dyn.rebal. station-based	179.1	1440	2.46	20.33	21.67
	Baseline 2 dyn.rebal. O-D	227.2	1440	2.44	19.79	21.33

The O-D variable model seems to provide slightly less lost demand, but at the cost of larger computing time due to increased model size.

6.3 Time discretization and time constraints

Table 24 shows the results of GT3 with different time-period lengths and time constraints. The conclusion is consistent with Table 7 in Section 5.3.

Table 24: Station-based model for GT3 with/without time constraints in 30/60 mins (60 stations, 4 trucks)

	Time Period (mins)	Time Constraints	O.F. Value	Time (mins)	MIP Gap (%)	Lost Demand (%)		Opt-sim-gap (%)	
						Rental	Return	Rental	Return
GT3	30	No	179.1	1440	2.46	20.33	21.67	16.56	49.23
	30	Yes	272.5	1440	4.69	19.29	20.57	12.58	38.29
	60	No	408.4	1440	1.00	21.28	22.65	11.57	34.98
	60	Yes	409.0	1440	1.73	21.66	22.40	12.00	35.28

We now explore the impacts of time constraints and time-period length for the O-D variable model.

The results for the model with O-D variables are summarized in Table 25. The conclusion is similar to the station-based trip variable model. However, when time constraints are applied, the running times for the models are much longer and the experiments for GT2 run out of memory.

We also carry out the same experiments on a 30 stations network. The results for them are shown in Table 26. Given that most of the stations can be reached within 30 minutes, time constraints are less effective. Compared to Table 7, a short time-period and time constraints are a good combination to guarantee enough rebalancing operations and timely arrivals for a larger studying area. However,

Table 25: O-D model with/without time constraints in 30/60 mins (60 stations, 4 trucks)

	Time Period (mins)	Time Constraints	O.F. Value	Time (mins)	MIP Gap (%)	Lost Demand (%)		Opt-sim-gap (%)	
						Rental	Return	Rental	Return
GT1	30	No	52.8	< 1	0.00	7.98	5.97	5.03	11.73
	30	Yes	53.0	465	0.00	7.95	6.43	4.95	12.18
	60	No	71.9	1440	0.09	8.40	4.74	4.69	9.94
	60	Yes	71.9	1440	0.10	8.44	4.03	4.75	9.17
GT3	30	No	227.2	1440	2.44	19.79	21.33	13.89	44.67
	30	Yes	328.0	1440	6.05	20.85	21.22	10.60	40.32
	60	No	494.3	1440	3.44	22.52	21.69	5.70	34.94
	60	Yes	496.0	1440	3.88	22.50	20.61	5.57	32.92

for small-scale BSSs, like 30 densely distributed stations, it is not worth applying time constraints, resulting in a marginal improvement and a long optimization time.

Table 26: Station-based variable model with/without time constraints in 30/60 mins (30 stations, 2 trucks)

	Time Period(mins)	Time Constraints	O.F. Value	Time (mins)	MIP Gap (%)	Lost Demand (%)		Opt-sim-gap (%)	
						Rental	Return	Rental	Return
GT1	30	No	0.5	< 1	0.00	8.95	4.73	9.83	11.80
	30	Yes	0.5	< 1	0.00	8.90	4.21	9.76	11.17
	60	No	5.3	< 1	0.00	9.12	2.55	9.07	9.76
	60	Yes	5.3	< 1	0.00	8.93	1.35	8.85	2.03

6.4 Trip distribution and variables domains for O-D model

The experiments for GT3 of the station-based model are carried out and demonstrated in Table 27. Although GT3 is hard to solve, Constraints (TD1) perform best in lost rental as in Table 8.

Table 27: Station-based model for GT3 with different trip distribution constraints (60 stations, 4 trucks, 30mins)

	Constraints	O.F. Value	Time (mins)	MIP Gap (%)	Lost Demand (%)		Opt-sim-gap (%)	
					Rental	Return	Rental	Return
GT3	(TD1)	875.8	1440	5.21	17.71	26.18	3.19	13.49
	(TD6)	205.7	1440	1.75	19.93	22.20	16.31	46.16
	(TD3)+(TD6)	304.0	1440	2.71	19.62	20.94	11.61	37.83
	(TD3)+(TD4)	373.7	1440	4.03	20.42	22.48	8.67	39.39
	None	179.1	1440	2.41	20.33	21.67	16.56	49.23

Concerning the model with O-D variables as shown in Table 28, adding trip distribution constraints is even harder to solve due to a large number of variables and constraints. Using Constraints (TD7) reduces both the lost rental and the opt-sim-gap. However, the running time is significantly longer.

Table 28: O-D model with different trip distribution constraints(60 stations, 4 trucks, 30mins)

	Constraints	O.F. Value	Time (mins)	MIP Gap (%)	Lost Demand (%)		Opt-sim-gap (%)	
					Rental	Return	Rental	Return
GT1	(TD7)	55.3	1440	4.90	6.86	6.50	3.65	10.90
	None	52.8	< 1	0.00	7.98	5.97	5.03	11.73
GT2	(TD7)	53.3	869	0.07	7.94	2.10	4.84	7.11
	None	51.5	288	0.00	9.95	2.27	7.24	9.75
GT3	(TD7)	248.7	1440	4.90	19.79	22.06	12.94	44.85
	None	227.2	1440	2.42	19.79	21.33	13.89	44.67

We finally present the results for trip distribution constraints with different variable domains for the O-D model in Table 29. Two instances out of five cannot be solved in the All-continuous model within 24 hours under constraints (TD7) for both the All-continuous model and the Partially-integer model. The Partially-integer model outperforms the All-continuous model with an obvious improvement in lost demand with or without (TD7). Since the MIP gap is really close to 0.01%, we can conclude that trip distribution constraints (TD7) give the best performance. The All-integer model for O-D variables is not considered here since its integer requirement for trip variables $x_{s,s'}^{t,t'}$ are too strict.

Table 29: O-D model with Different variable domains and trip distribution constraints for GT1 (30 stations, 2 trucks, 30 mins)

Variable Domain	Const-raints	O.F. Value	Time (mins)	MIP Gap (%)	Lost Demand (%)		Opt-sim-gap (%)	
					Rental	Return	Rental	Return
All-continuous	(TD7)	23.0	585	0.03	8.37	4.11	5.37	9.89
	None	22.7	288	0.00	10.07	4.60	7.39	12.58
Partially-integer	(TD7)	23.0	864	0.03	4.12	1.40	0.72	2.16
	None	22.7	288	0.00	7.26	4.40	4.12	8.92

6.5 Sequences of events

The results of event sequences for GT3 are shown in Table 30. Compared to Table 10, sequences (d)(r)(a), (r)(d)(a), and (d)(a)(r) are still perform well.

Table 30: Station-based model with different sequences of events (60 stations, 4 trucks, 30 mins)

	Sequences of events	O.F. Value	Time (mins)	MIP Gap (%)	Lost Demand(%)		Opt-sim-gap(%)	
					Rental	Return	Rental	Return
GT3	(r)(a)(d)	246.9	1440	2.35	19.57	21.98	15.14	41.38
	(a)(d)(r)	295.4	1440	2.32	20.59	20.63	14.24	39.82
	(d)(r)(a)	302.8	1440	3.22	19.22	21.70	8.94	41.28
	(r)(d)(a)	280.3	1440	2.94	19.60	20.68	11.93	39.47
	(d)(a)(r)	318.8	1440	3.12	19.23	19.74	8.57	36.62
	(a)(r)(d)	259.6	1440	1.91	19.45	21.89	15.08	39.42
	(r)(a+d)	236.7	1440	2.39	19.41	22.60	15.27	42.86
	(a+d)(r)	285.0	1440	2.24	19.87	19.70	13.46	38.00

References

[1] Jan Brinkmann, Marlin W Ulmer, and Dirk C Mattfeld. Dynamic lookahead policies for stochastic-dynamic inventory routing in bike sharing systems. Computers & Operations Research, 106:260–279, 2019.

[2] Jan Brinkmann, Marlin W Ulmer, and Dirk C Mattfeld. The multi-vehicle stochastic-dynamic inventory routing problem for bike sharing systems. Business Research, 13(1):69–92, 2020.

[3] Christine Fricker and Nicolas Gast. Incentives and redistribution in homogeneous bike-sharing systems with stations of finite capacity. Euro journal on transportation and logistics, 5(3):261–291, 2016.

[4] Benjamin Legros. 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, 2019.

[5] Xue Luo, Li Li, Lei Zhao, and Jianfeng Lin. Dynamic intra-cell repositioning in free-floating bike-sharing systems using approximate dynamic programming. Transportation Science, 2022.

[6] Young-Hyun Seo, Dong-Kyu Kim, Seungmo Kang, Young-Ji Byon, and Seung-Young Kho. Rebalancing docked bicycle sharing system with approximate dynamic programming and reinforcement learning. Journal of Advanced Transportation, 2022, 2022.

[7] Ian Xiao. A distributed reinforcement learning solution with knowledge transfer capability for a bike rebalancing problem. arXiv preprint arXiv:1810.04058, 2018.

[8] Claudio Contardo, Catherine Morency, and Louis-Martin Rousseau. Balancing a dynamic public bike-sharing system, volume 4. Cirrelt Montreal, Canada, 2012.

[9] Supriyo Ghosh, Pradeep Varakantham, Yossiri Adulyasak, and Patrick Jaillet. Dynamic repositioning to reduce lost demand in bike sharing systems. Journal of Artificial Intelligence Research, 58:387–430, 2017.

[10] Christian Kloimüllner, Petrina Papazek, Bin Hu, and Günther R Raidl. Balancing bicycle sharing systems: an approach for the dynamic case. In European Conference on Evolutionary Computation in Combinatorial Optimization, pages 73–84. Springer, 2014.

[11] Mauro Dell'Amico, Manuel Iori, Stefano Novellani, and Anand Subramanian. The bike sharing rebalancing problem with stochastic demands. Transportation research part B: methodological, 118:362–380, 2018.

[12] Chenyi Fu, Ning Zhu, Shoufeng Ma, and Ronghui Liu. A two-stage robust approach to integrated station location and rebalancing vehicle service design in bike-sharing systems. European Journal of Operational Research, 298(3):915–938, 2022.

[13] Supriyo Ghosh, Michael Trick, and Pradeep Varakantham. Robust repositioning to counter unpredictable demand in bike sharing systems. In Proceedings of the 25th International Joint Conference on Artificial Intelligence IJCAI, pages 3096–3102, 2016.

[14] Meghna Lowalekar, Pradeep Varakantham, Supriyo Ghosh, Sanjay Dominik Jena, and Patrick Jaillet. Online repositioning in bike sharing systems. In Twenty-Seventh International Conference on Automated Planning and Scheduling, 2017.

[15] Carlos M Vallez, Mario Castro, and David Contreras. Challenges and opportunities in dock-based bike-sharing rebalancing: a systematic review. Sustainability, 13(4):1829, 2021.

[16] Zulqarnain Haider, Alexander Nikolaev, Jee Eun Kang, and Changhyun Kwon. Inventory rebalancing through pricing in public bike sharing systems. European Journal of Operational Research, 270(1):103–117, 2018.

[17] Zhenghua Hu, Kejie Huang, Enyou Zhang, Qi'ang Ge, and Xiaoxue Yang. Rebalancing strategy for bike-sharing systems based on the model of level of detail. Journal of advanced transportation, 2021, 2021.

[18] The meddin bike-sharing world map report, 2022. https://bikesharingworldmap.com/.

[19] Yuchuan Du, Fuwen Deng, and Feixiong Liao. A model framework for discovering the spatio-temporal usage patterns of public free-floating bike-sharing system. Transportation Research Part C: Emerging Technologies, 103:39–55, 2019.

[20] Chengcheng Xu, Junyi Ji, and Pan Liu. The station-free sharing bike demand forecasting with a deep learning approach and large-scale datasets. Transportation research part C: emerging technologies, 95:47–60, 2018.

[21] Xiaojian Zhang, Hongtai Yang, Rong Zheng, Zhicheng Jin, and Bowen Zhou. A dynamic shared bikes rebalancing method based on demand prediction. In 2019 IEEE Intelligent Transportation Systems Conference (ITSC), pages 238–244. IEEE, 2019.

[22] Aritra Pal and Yu Zhang. Free-floating bike sharing: Solving real-life large-scale static rebalancing problems. Transportation Research Part C: Emerging Technologies, 80:92–116, 2017.

[23] Tal Raviv, Michal Tzur, and Iris A Forma. Static repositioning in a bike-sharing system: models and solution approaches. EURO Journal on Transportation and Logistics, 2(3):187–229, 2013.

[24] Hans Martin Espegren, Johannes Kristianslund, Henrik Andersson, and Kjetil Fagerholt. The static bicycle repositioning problem-literature survey and new formulation. In International Conference on Computational Logistics, pages 337–351. Springer, 2016.

[25] CS Shui and WY Szeto. A review of bicycle-sharing service planning problems. Transportation Research Part C: Emerging Technologies, 117:102648, 2020.

[26] Federico Chiariotti, Chiara Pielli, Andrea Zanella, and Michele Zorzi. A dynamic approach to rebalancing bike-sharing systems. Sensors, 18(2):512, 2018.

[27] Xinghua Zheng, Ming Tang, Yuechang Liu, Zhengzheng Xian, and Hankz Hankui Zhuo. Repositioning bikes with carrier vehicles and bike trailers in bike sharing systems. Applied Sciences, 11(16):7227, 2021.

[28] Kiana Roshan Zamir. Dynamic Repositioning For Bikesharing Systems. PhD thesis, 2020.

[29] Jia Shu, Mabel C Chou, Qizhang Liu, Chung-Piaw Teo, and I-Lin Wang. Models for effective deployment and redistribution of bicycles within public bicycle-sharing systems. Operations Research, 61(6):1346–1359, 2013.

[30] Supriyo Ghosh, Jing Yu Koh, and Patrick Jaillet. Improving customer satisfaction in bike sharing systems through dynamic repositioning. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19, pages 5864–5870, 2019.

[31] Jie Zhang, Meng Meng, Yiik Diew Wong, Petros Ieromonachou, and David ZW Wang. A data-driven dynamic repositioning model in bicycle-sharing systems. International Journal of Production Economics, 231:107909, 2021.

[32] Dong Zhang, Chuhang Yu, Jitamitra Desai, HYK Lau, and Sandeep Srivathsan. A time-space network flow approach to dynamic repositioning in bicycle sharing systems. Transportation research part B: methodological, 103:188–207, 2017.

[33] Peng-Sheng You. A two-phase heuristic approach to the bike repositioning problem. Applied Mathematical Modelling, 73:651–667, 2019.

[34] Federico Chiariotti, Chiara Pielli, Andrea Zanella, and Michele Zorzi. A bike-sharing optimization framework combining dynamic rebalancing and user incentives. ACM Transactions on Autonomous and Adaptive Systems (TAAS), 14(3):1–30, 2020.

[35] Giuseppe C Calafiore, Christian Bongiorno, and Alessandro Rizzo. A robust mpc approach for the rebalancing of mobility on demand systems. Control Engineering Practice, 90:169–181, 2019.

[36] Konstantina Mellou and Patrick Jaillet. Dynamic resource redistribution and demand estimation: An application to bike sharing systems. Available at SSRN 3336416, 2019.

[37] Meng Yuan, Qiong Zhang, Bohong Wang, Yongtu Liang, and Haoran Zhang. A mixed integer linear programming model for optimal planning of bicycle sharing systems: A case study in beijing. Sustainable cities and society, 47:101515, 2019.

[38] Bixi open data, 2023. https://bixi.com/en/open-data-2/.

[39] Iris A Forma, Tal Raviv, and Michal Tzur. A 3-step math heuristic for the static repositioning problem in bike-sharing systems. Transportation research part B: methodological, 71:230–247, 2015.

[40] Supriyo Ghosh, Pradeep Varakantham, Yossiri Adulyasak, and Patrick Jaillet. Dynamic redeployment to counter congestion or starvation in vehicle sharing systems. In Proceedings of the International Conference on Automated Planning and Scheduling, volume 25, pages 79–87, 2015.

[41] Jianbin Huang, Qinglin Tan, He Li, Ao Li, and Longji Huang. Monte carlo tree search for dynamic bike repositioning in bike-sharing systems. Applied Intelligence, pages 1–16, 2022.

[42] Yu Jin, Cesar Ruiz, and Haitao Liao. A simulation framework for optimizing bike rebalancing and maintenance in large-scale bike-sharing systems. Simulation Modelling Practice and Theory, 115:102422, 2022.

[43] Junming Liu, Leilei Sun, Weiwei Chen, and Hui Xiong. Rebalancing bike sharing systems: A multi-source data smart optimization. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, pages 1005–1014, 2016.

[44] A set of bixi stations, google maps, google, 2023. Accessed on August 2023.

[45] S Mohammad Arabzad, Hadi Shirouyehzad, Mahdi Bashiri, Reza Tavakkoli-Moghaddam, and Esmaeil Najafi. Rebalancing static bike-sharing systems: a two-period two-commodity multi-depot mathematical model. Transport, 33(3):718–726, 2018.

[46] Hazal Akova, Selin Hulagu, and Hilmi Berk Celikoglu. Static bike repositioning problem with heterogeneous distribution characteristics in bike sharing systems. Transportation Research Procedia, 62:205–212, 2022.

[47] Mike Benchimol, Pascal Benchimol, Benoît Chappert, Arnaud De La Taille, Fabien Laroche, Frédéric Meunier, and Ludovic Robinet. Balancing the stations of a self service “bike hire” system. RAIRO-Operations Research, 45(1):37–61, 2011.

[48] Bruno P Bruck, Fábio Cruz, Manuel Iori, and Anand Subramanian. The static bike sharing rebalancing problem with forbidden temporary operations. Transportation science, 53(3):882–896, 2019.

[49] Daniel Chemla, Frédéric Meunier, and Roberto Wolfler Calvo. Bike sharing systems: Solving the static rebalancing problem. Discrete Optimization, 10(2):120–146, 2013.

[50] Jiming Chen, Zidong Yang, Peng Cheng, and Yuanchao Shu. Rebalancing bike-sharing system with deep sequential learning. IEEE Intelligent Transportation Systems Magazine, 13(4):92–98, 2020.

[51] Mauro Dell'Amico, Eleni Hadjicostantinou, Manuel Iori, and Stefano Novellani. The bike sharing rebalancing problem: Mathematical formulations and benchmark instances. Omega, 45:7–19, 2014.

[52] Luca Di Gaspero, Andrea Rendl, and Tommaso Urli. A hybrid aco+ cp for balancing bicycle sharing systems. In International Workshop on Hybrid Metaheuristics, pages 198–212. Springer, 2013.

[53] Güneş Erdoğan, Gilbert Laporte, and Roberto Wolfler Calvo. The static bicycle relocation problem with demand intervals. European Journal of Operational Research, 238(2):451–457, 2014.

[54] Güneş Erdoğan, Maria Battarra, and Roberto Wolfler Calvo. An exact algorithm for the static rebalancing problem arising in bicycle sharing systems. European Journal of Operational Research, 245(3):667–679, 2015.

[55] Christian Kloimüllner and Günther R Raidl. Full-load route planning for balancing bike sharing systems by logic-based benders decomposition. Networks, 69(3):270–289, 2017.

[56] Bahman Lahoorpoor, Hamed Faroqi, Abolghasem Sadeghi-Niaraki, and Soo-Mi Choi. Spatial cluster-based model for static rebalancing bike sharing problem. Sustainability, 11(11):3205, 2019.

[57] Yanfeng Li, WY Szeto, Jiancheng Long, and CS Shui. A multiple type bike repositioning problem. Transportation Research Part B: Methodological, 90:263–278, 2016.

[58] Marian Rainer-Harbach, Petrina Papazek, Bin Hu, and Günther R Raidl. Balancing bicycle sharing systems: A variable neighborhood search approach. In European conference on evolutionary computation in combinatorial optimization, pages 121–132. Springer, 2013.

[59] Marian Rainer-Harbach, Petrina Papazek, Günther R Raidl, Bin Hu, and Christian Kloimüllner. Pilot, grasp, and vns approaches for the static balancing of bicycle sharing systems. Journal of Global Optimization, 63(3):597–629, 2015.

[60] Jasper Schuijbroek, Robert C Hampshire, and W-J Van Hoeve. Inventory rebalancing and vehicle routing in bike sharing systems. European Journal of Operational Research, 257(3):992–1004, 2017.

[61] WY Szeto, Ying Liu, and Sin C Ho. Chemical reaction optimization for solving a static bike repositioning problem. Transportation research part D: transport and environment, 47:104–135, 2016.

[62] Qiong Tang, Zhuo Fu, Dezhi Zhang, Hao Guo, and Minyi Li. Addressing the bike repositioning problem in bike sharing system: A two-stage stochastic programming model. Scientific Programming, 2020, 2020.

[63] Bowen Zhang, Xiang Li, and Francisco Saldanha-da Gama. Free-floating bike-sharing systems: New repositioning rules, optimization models and solution algorithms. Information Sciences, 600:239–262, 2022.

[64] Marte D Gleditsch, Kristine Hagen, Henrik Andersson, Steffen J Bakker, and Kjetil Fagerholt. A column generation heuristic for the dynamic bicycle rebalancing problem. European Journal of Operational Research, 2022.

[65] Runqiu Hu, Zhizheng Zhang, Xinwei Ma, and Yuchuan Jin. Dynamic rebalancing optimization for bike-sharing system using priority-based moea/d algorithm. IEEE Access, 9:27067–27084, 2021.

[66] CS Shui and WY Szeto. Dynamic green bike repositioning problem–a hybrid rolling horizon artificial bee colony algorithm approach. Transportation Research Part D: Transport and Environment, 60:119–136, 2018.

[67] Akihiro Yoshida, Yosuke Yatsushiro, Nozomi Hata, Tatsuru Higurashi, Nariaki Tateiwa, Takashi Wakamatsu, Akira Tanaka, Kenichi Nagamatsu, and Katsuki Fujisawa. Practical end-to-end repositioning algorithm for managing bike-sharing system. In 2019 IEEE International Conference on Big Data (Big Data), pages 1251–1258. IEEE, 2019.