CIRRELT-2023-37

Dynamic Rebalancing for Bike-sharing Systems under Inventory Interval and Target Predictions

Jiaqi Liang
Maria Clara Martins Silva
Daniel Aloise
Sanjay Dominik Jena

November 2023

Bureau de Montréal
Université de Montréal
C.P. 6128, succ. Centre-Ville
Montréal (Québec) H3C 3J7
Tél : 1-514-343-7575
Télécopie : 1-514-343-7121

Bureau de Québec
Université Laval,
2325, rue de la Terrasse
Pavillon Palasis-Prince, local 2415
Québec (Québec) G1V 0A6
Tél : 1-418-656-2073
Télécopie : 1-418-656-2624

Dynamic Rebalancing for Bike-sharing Systems under Inventory Interval and Target Predictions

Jiaqi Liang[1,2], Maria Clara Martins Silva[3], Daniel Aloise[3], Sanjay Dominik Jena[1,4,*]

[1] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)

[2] Department of Mathematics and Industrial Engineering, Polytechnique Montréal

[3] Computer and Software Engineering Department, Polytechnique Montréal

[4] Analytics, Operations, and Information Technologies Department, School of Management, Université du Québec à Montréal

Abstract. Bike-sharing systems have been implemented in multiple major cities, offering a low-cost and environmentally friendly transportation alternative to vehicles. Due to the stochastic nature of customer trips, stations are often unbalanced, resulting in unsatisfied demand. As a remedy, operators employ trucks to rebalance bikes among unbalanced stations. Given the complexity of the dynamic rebalancing planning, this topic has received significant attention from the Operations Research community. However, the planning problem requires significant simplifications such that optimization models remain computationally tractable. 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. Indeed, existing simulation models to evaluate the performance of planning strategies also rely on simplifications, such as the aggregation of trips within the same time-period, therefore ignoring the original chronological sequence of trip demand. In this paper, we first systematically survey the literature on rebalancing problems and their modeling assumptions. We then propose a general modeling framework for multi-period dynamic 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 more realistic simulator to evaluate the operational performance of rebalancing strategies. Finally, extensive numerical experiments are carried out to analyze the effectiveness of various modeling assumptions and techniques. In this way, we identify the assumptions that empirically provide the most effective rebalancing strategies in practice. In particular, a set of specific trip distribution constraints as well as event sequences ignored in the previous literature seem to provide particularly good results.

Keywords: 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 first author was a Ph.D. student at Polytechnique Montreal in the Canada Excellence Research Chair "Data Science for Real-time Decision-making". The work of the second author was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada under grant 2017-05224.

Results and views expressed in this publication are the sole responsibility of the authors and do not necessarily reflect those of CIRRELT.

Les résultats et opinions contenus dans cette publication ne reflètent pas nécessairement la position du CIRRELT et n'engagent pas sa responsabilité.

* Corresponding author: jena.sanjay-dominik@uqam.ca

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

1. Introduction

The pursuit of environmentally friendly transportation modes has increased considerably in the last few years, with Bike-Sharing systems (BSS) having emerged as a notable choice. As of 2022, there were over 1,900 BSSs in operation comprising almost 9 million bikes [43], offering cities opportunities to reduce carbon emissions, traffic relief, and improve the quality of life for their residents [9].

A particular challenge in the management of BSSs are station imbalances. Most users follow consistent travel patterns, often commuting toward commercial areas (such as city centers) during morning peak hours and returning to residential zones after work. Such behavior often results in stations being full or empty, leading to *lost demand*, i.e., rental demand that cannot be met due to an empty station or return demand that cannot be met as a result of a full station. Occasional user trips introduce stochasticity, further aggravating station imbalances. As a remedy, BSS operators often redistribute bikes among the stations, typically, via trucks, a process known as *rebalancing*. Rebalancing imbalanced stations has been proven to be more cost-effective than alternative solutions, such as adding more stations or installing additional docks [53]. The development of effective rebalancing strategies has therefore become a crucial research field with the potential to significantly improve user satisfaction. Two primary rebalancing schemes have been acknowledged in the literature: overnight station rebalancing and intraday rebalancing [44; 48]. The latter is often referred to as the Dynamic Bicycle Repositioning Problem (DBRP)[32; 42]. In contrast to overnight rebalancing, the DBRP involves continuous intraday rebalancing operations in parallel to the user trips occuring throughout the day. Given its higher impact on demand satisfaction, we here focus on this problem.

While a few recent works use Markov Decision Processes [5; 33; 34; 35; 52], the majority of the literature applies Mixed-integer Programming (MIP) models [23; 42; 63; 65], given that they are flexible and widely used within industrial decision-making processes. Among MIP models, multi-period models benefit from an integrated planning over all time-periods and do therefore not suffer from the myopic behavior of single-period models.

Most models aim at minimizing lost demand or maximizing successful trips [23; 39; 54; 61; 63; 64], with both rental and return demands estimated either by naive predictions (e.g., the historical mean) [20; 22; 39; 63] or more sophisticated Machine Learning (ML) techniques [62; 64]. However, minimizing lost demand may result in sub-optimal rebalancing solutions if the predicted rental and return are not accurate enough. BSS operators, like BIXI Montreal, often use *inventory intervals* and *target inventories* within their rebalancing planning. The inventory interval of a station refers to the acceptable range of its inventory, ensuring that the station maintains a level of available bikes and free docks. The target inventory of a station represents the desired number of bikes at that station to ensure optimal service.

While many methods have been proposed to compute inventory intervals and target inventories (see e.g., [26; 28; 38; 47]), only a few have incorporated them into optimization models. Notably, most relevant studies [24; 32; 51] either focus on single-period models or minimize the deviation of target inventory at the end of the planning process only. Although [56] and [58] attempt to introduce intervals into multi-period models, they relax the intervals by using station capacities directly in their experiments. Furthermore, the intervals and targets in these models are often determined without considering weather conditions, which significantly impact user behavior. As a result, the benefits of combining optimization algorithms with intervals or targets in the objective function for multi-period models are still to be determined.

In a similar vein, while some of the existing models have been carried out in a rolling fashion (see, e.g., [23; 24; 42; 62; 54]), literature has not yet quantified the benefits of integrating system status update through reoptimization (i.e., rolling and folding planning) over classical static planning of multi-period rebalancing models. In this paper, we aim at filling these gaps and approach the DBRP by incorporating inventory intervals and target inventories.

Contributions. Our work evaluates the entire pipeline required for an automatized and data-driven rebalancing process in BSSs. The main contributions can be summarized as follows. (i) We propose two optimization models that integrate inventory intervals and target inventories into the objective functions, concepts that are often already used within the decision-making process of BSS operators. In contrast to classical models that minimize unmet demand, the proposed models tend to ensure a buffer in the station inventories and are therefore more capable of dealing with demand fluctuations. (ii) We propose a realistic instance generator, generating varying weather conditions for different days along with trip data that is historically coherent with such conditions. (iii) We conduct an extensive comparison among three multi-period models with different objective functions for DBRP, including the classical objective that minimizes unmet demand and the two proposed objectives that minimize the deviations from inventory intervals and target inventories. Demand predictions are obtained from an advanced machine learning model, capable of making sufficiently accurate predictions based on weather and temporal features. Inventory and target inventories are computed such that they maximize the desired service-level. The performance is estimated by a fine-grained discrete-event simulator. Our models demonstrate a remarkable robustness to cope with trip fluctuations, reducing lost demand by up to 34% as compared to the model minimizing lost demand. (iv) We empirically compare the impact of employing different reoptimization modes (i.e., static and rolling planning) for all models. The results indicate a clear advantage of reoptimizing over the planning horizon, reducing the lost demand by at least 30% on average, without necessarily increasing the computing time. (iv) We compare the impact of using perfect information and less accurate demand predictions on the performance of the planning models. Interestingly, our proposed optimization models remain remarkably robust. (v) A case-study on real-world data is considered, confirming the benefits of the proposed approaches.

Outline. This paper is organized as follows. Section 2 reviews relevant literature on the objective functions used in rebalancing models, inventory intervals and target inventories, trip prediction, and reoptimization modes. Section 3 reviews the baseline model that minimizes unmet demand and introduces two dynamic rebalancing models minimizing the deviations from inventory intervals and target inventories. Numerical experiments and analysis on synthetic and real-world data are presented in Section 4. This is followed by the conclusions in Section 5.

2. Literature Review for Rebalancing Problems in BSSs

This section reviews the literature related to the here considered planning problem and our contributions, focusing on the objective functions used within DBRP models, inventory intervals and targets, demand prediction, and reoptimization modes.

2.1. Objective functions in rebalancing models

We first review existing MIP models and their objective functions used in dynamic rebalancing. Metrics used in the objective functions can be classified into three different types (see Appendix of [36]): distance-based metrics, loading-based metrics, and demand-based metrics. Distance-based metrics are associated with the travelling distance of vehicles, mainly including travelling costs, travelling time, and fuel consumption (see e.g., [1; 22; 46; 65]). Loading-based metrics are associated with the number of handling (i.e., loading and unloading) operations (see e.g., [25; 55]). Handling cost or time reflects the workload of operations. Finally, demand-based metrics concern the dissatisfaction of customers. Some studies consider more than one aspect in their objective functions (see, e.g., [20; 25; 32; 42; 64]).

We here focus on demand-based metrics, which have been more relevant in the literature, and to which our contributions are directly related. The most common approach for demand-based metrics is to minimize the lost rental and return demand, which is also equivalent to maximizing successful trips (see e.g., [10; 22; 23; 25; 39; 54; 61; 63; 64; 65]). Concurrently, there have been a few attempts to minimize the deviation between the inventories of the stations and their target inventories [24; 32] or inventory intervals [58], which still holds considerable potential for further exploration. Our work focuses on such inventory intervals and target inventories, for which the related literature is reviewed next.

2.2. Inventory intervals and target inventories

As opposed to minimizing lost demand, the concepts of inventory intervals and target inventories have been found to be useful within the rebalancing decision-making process. Inventory intervals define an acceptable range of the bike inventory at individual stations, whereas target inventories represent specific inventory levels that operators aim to uphold at each station. Both are typically designed to ensure a high rate of demand satisfaction.

Even though inventory intervals and target inventories are often used concepts in the planning processes of BSS operators, only a few works have incorporated them into optimization models. When using target inventories, objective functions typically minimize the deviations between the station inventories and the specified targets. Here, [24] consider a single-period model, executed in a rolling planning, that minimizes such inventory deviations. Further, [32] propose a multi-period rebalancing model for which they compare various metaheuristics. Next to multiple criteria (such as minimizing costs) considered within the objective function, the model also aims at aligning the final station inventory at the end of the planning horizon with a specific target inventory. [8] define target values for entire city zones to ensure a more equitable rebalancing process in which remote areas are not neglected due to their lower population (and demand) density. Finally, the model introduced by [7] also uses target values, but instead of minimizing deviations, they use hard constraints to ensure that station inventories equal those targets at the end of the rebalancing process. As a result, the minimization of the deviations between station inventories and target values at each of the time-periods, as a mean of improving demand satisfaction, has not yet been considered and deserves further investigation.

Inventory intervals have also been implemented into rebalancing models as hard constraints [51; 56], requiring that station inventories remain within a specific range. Here, [51] focus on the static rebalancing problem with a single time-period, while [56] propose a multi-period model. However, the authors do not consider these constraints in their computational experiments. Given that hard constraints may easily lead to infeasible outcomes, [58] present a multi-period model that, among several other criteria, minimizes the deviation of station inventories from predefined inventory intervals within the objective function. Unfortunately, their experimental setup also ignores such intervals, leaving the actual effectiveness of inventory intervals within the optimization model unexplored.

While the works cited above assume that target inventories and inventory intervals are given, a few works also propose how to effectively compute them. Target inventories have often been computed such that they reduce the probability of a station reaching both the full and empty status, typically requiring the prior estimation of rental and return distribution on historical data [18; 24; 26; 47]. Inventory intervals have been computed in a similar fashion [51; 28], selecting those that minimize the likelihood of a station becoming either empty or full. Finally, different from those approaches, [11] and [29] simulate the performance of several sets of initial inventories and select the one that performs best.

Most works cited above compute target inventories and inventory intervals based on historical trip demand. However, they disregard external factors such as weather conditions, the importance of which has been widely acknowledged in the literature (see, e.g., [13; 17; 19; 28; 31; 38; 43]). To this end, [28] extend the notion of service-levels proposed by [51], computing both target inventories and inventory intervals based on the service level and the predicted demand. The latter is estimated based on machine learning models trained on both temporal and weather data, therefore holding the potential of providing better performing target inventories and inventory intervals. The authors also introduce two additional hyperparameters, α and β, allowing operators to align inventory intervals and target inventories with their priorities for either rentals or returns (see Appendix A for more details), hence making it an attractive approach to operators.

2.3. Demand prediction for BSSs

Demand prediction is an essential step in the rebalancing process, enabling operators to anticipate which stations require higher inventories to better serve trip demand. Accurately predicting rentals and returns is challenging, as it is influenced by numerous factors [57]. Literature on demand prediction in BSSs can be divided into approaches predicting at global demand level (see e.g., [19; 49; 60]), at the level of station-clusters (see e.g., [3; 14; 57]), and at the level of individual stations (see e.g., [4; 12; 28; 45; 62]). We here focus on works predicting demand at station level, which is required for rebalancing operations since they are tailored considering the rentals and returns for each station.

Different techniques have been used to predict demand in BSSs. The average of historical trips (i.e., rentals and returns) can be used to estimate future demand [2; 11; 20; 24; 47; 51; 63], which can be seen as a naive predictor. Alternatively, rental and

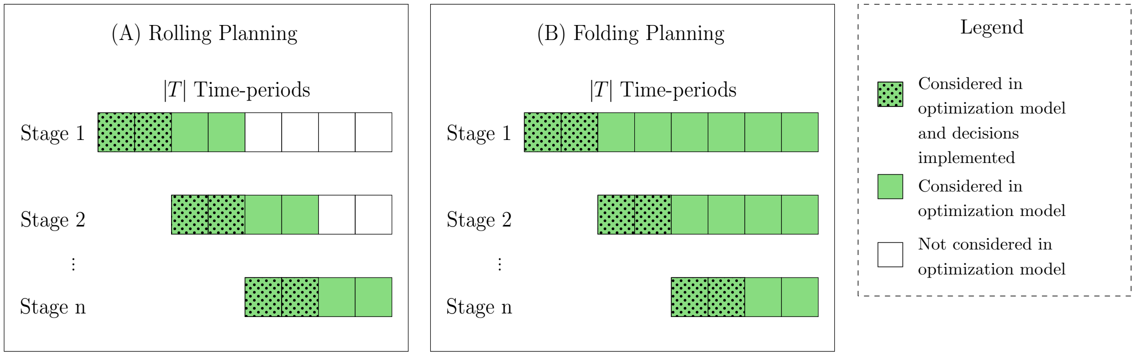
(A) Rolling Planning
|T| Time-periods
Stage 1
Stage 2
⋮
Stage n
(B) Folding Planning
|T| Time-periods
Stage 1
Stage 2
⋮
Stage n
Legend
Considered in optimization model and decisions implemented
Considered in optimization model
Not considered in optimization model

Figure 1: The structure of rolling and folding planing

return (often Poisson) distributions have been estimated from historical trip data, from which trip demand is then sampled and used within the optimization models (see, e.g., [22; 39; 63; 65]). While such prediction methods centered on historical mean demand have been quite popular, several studies (see, e.g., [16; 38; 40] haven suggested that such methods may result in high prediction errors when compared to ML models, as the latter can take into considerations features beyond historical trip data, including weather conditions, time of day, day of the week and the occurrence of special events.

Generally, ML algorithms can capture intricate patterns and correlations and may, therefore, result in significantly more accurate demand predictions. Random forests and gradient boosted trees, in particular, have been found to provide competitive prediction accuracy in the context of rental and return predictions [4; 28; 40; 59; 60]. Typically, such models integrate weather and temporal features, highlighting their importance to accurately predict demand. Here, [28] utilize Singular Value Decomposition, a dimension reduction technique, to reduce the dimensionality of the trip data, eliminate noise and improve both time required and the accuracy of hourly station-level rental and return predictions (details can be found in Appendix A). These reasons make this model an attractive option to estimate the future trip demand for our rebalancing optimization models.

2.4. Reoptimization modes: static, rolling, and folding

In practice, both single-period and multi-period models can be implemented in several ways. A simple, yet common approach (see, e.g., [20; 32; 41; 50; 53; 64]) is to optimize once over the entire planning horizon (i.e., all considered time-periods) and then implement all rebalancing decisions (i.e., the number of bikes dropped off and picked up at each station) as planned for all time periods.

While multi-period models can represent the consequences of decisions made at early time-periods, when executed within a static planning, they do not benefit from updated system information (such as station inventories or improved demand predictions). Therefore, practitioners often tend to reoptimize the rebalancing decisions throughout the planning horizon. Specifically, rolling planning (also called rolling window planning) considers the reoptimization over several time-periods at predefined reoptimization stages. Figure 1 (A) depicts the rolling planning, where, at each reoptimization stage, the green squares indicate time periods considered in the optimization models, and the green dotted squares indicate decisions of the time-periods actually executed in practice. This approach has been popular, with works implementing rolling planning for both multi-period (see, e.g., [23; 39; 42; 62; 54; 63]) and single-period models (see e.g., [21; 24; 25]). This allows for correcting ineffective planning, e.g., due to forecasting inaccuracies. In addition, considering only a subset of all time-periods within the planning window has the advantage of resulting in a more tractable optimization model.

If model decisions at early time-periods may impact decisions several time-periods ahead, one may want to consider a model with a longer planning horizon. Folding planning therefore optimizes on the remaining planning horizon, as depicted in Figure 1 (B). While this is generally a common approach in multi-period models [37], to the best of our knowledge, this approach has not yet been explored for BSS rebalancing planning.

3. Dynamic Rebalancing Models

In this section, we formulate the DBRP as MIP models. Section 3.1 first describes a multi-period model, which minimizes the unmet rental and return demand, the more popular objective in the literature (as discussed in Section 2.1). We then propose two models integrating inventory intervals and target inventories into the objective functions in Section 3.2.

3.1. Multi-period rebalancing model minimizing lost demand

We consider the model with multiple time-periods from [36]. The input parameters are listed in Table 1 and decision variables are shown in Table 2. We denote S as the set of stations, while V denotes the set of available vehicles. Each station $s \in S$ has a capacity of C_s docks and each vehicle $v \in V$ can hold at most $\hat{C}_v$ bikes. We consider a planning horizon with $|T|$ time-periods, where each time period $t \in T$ represents a duration of L_t minutes.

We assume that a vehicle can visit only one station per time-period. As a result, a vehicle can visit at most T stations during the entire planning horizon. The decision

Table 1: Input parameters of the optimization model

Input Parameters	Definition
S	The set of stations.
V	The set of vehicles.
T	The set of discreted time-periods.
C_s	The capacity of station $s \in S$.
$\hat{C}_v$	The capacity of vehicle $v \in V$.
L_t	The duration (in minutes) of time period $t \in T$.
d_s^1	The initial number of bikes at the station $s \in S$.
$\hat{d}_v^1$	The initial number of bikes in vehicle $v \in V$.
$z_{s,v}^1$	The initial location of each vehicle $v \in V, s \in S$.
$f_s^{+,t}$	The expected rental demand at station $s \in S$ in period $t \in T$.
$f_s^{-,t}$	The expected return demand at station $s \in S$ in period $t \in T$.

Table 2: Decision variables of the optimization model

Variables	Definition
d_s^t	The number of bikes available at station $s \in S$ at the beginning of period $t \in T$.
$\hat{d}_v^t$	The number of bikes in vehicle $v \in V$ at the beginning of period $t \in T$.
$x_s^{+,t}$	The number of successful rentals starting from station $s \in S$ in period $t \in T$.
$x_s^{-,t}$	The number of successful returns ending at station $s \in S$ in period $t \in T$.
$r_{s,v}^{+,t}$	The number of bikes picked up at station $s \in S$ by vehicle $v \in V$ in period $t \in T$.
$r_{s,v}^{-,t}$	The number of bikes dropped off at station $s \in S$ by vehicle $v \in V$ in period $t \in T$.
$z_{s,v}^t$	$z_{s,v}^t = 1$, if vehicle $v \in V$ visits station $s \in S$ in period $t \in T$; 0 otherwise.

variables $r_{s,v}^{+,t}$ and $r_{s,v}^{-,t}$ represent the number of bikes vehicle v picks up and drops off, respectively, at station s during period t. Furthermore, binary variable $z_{s,v}^t$ takes value 1 if and only if vehicle v visits station s at time-period t. For each time-period, intermediate variables are used: the number of bikes available at stations and in vehicles, successful trips, and vehicle routes. The resulting MIP model is expressed as follows:

The objective function (1) minimizes the total lost demand, i.e., the unmet expected demand for both rentals and returns, over the entire planning horizon at all stations. Constraints (2) compute the number of bikes in each vehicle v in period $t+1$ based on the number of bikes in the previous period and the number of picked up/ dropped off bikes. Constraints (3) compute the number of bikes in period $t+1$ at each station s as the sum of the number of bikes of that station in the previous period, the number of bikes rebalanced by vehicles, and those moved by users (i.e., successful rentals and returns). Constraints (4) ensure that each vehicle v can only be at one station at each time-period. Constraints (5) ensure that a vehicle can perform operations at a station only when it is present at that station. Constraints (6) impose that the number of bikes in each vehicle is bounded by its capacity. Constraints (7) are the capacity constraints for the stations. Constraints (8) bound the number of successful trips by the expected rental and return. Finally, constraints (9) enforce that the pick-up and drop-off operations respect the vehicle's capacities.

$$
\begin{aligned}
\min \quad & \sum_{s\in S}\sum_{t\in T}(f_s^{+,t} - x_s^{+,t}) + \sum_{s\in S}\sum_{t\in T}(f_s^{-,t} - x_s^{-,t}) && && (1)\\
\text{s.t.} \quad & \hat{d}_v^{t+1} = \hat{d}_v^t + \sum_{s\in S}(r_{s,v}^{+,t} - r_{s,v}^{-,t}) && \forall\ v\in V,\ t\in T && (2)\\
& d_s^{t+1} = d_s^t - \sum_{v\in V}(r_{s,v}^{+,t} - r_{s,v}^{-,t}) - x_s^{+,t} + x_s^{-,t} && \forall\ s\in S,\ t\in T && (3)\\
& \sum_{s\in S} z_{s,v}^t = 1 && \forall\ v\in V,\ t\in T && (4)\\
& r_{s,v}^{+,t} + r_{s,v}^{-,t} \le \hat{C}_v z_{s,v}^t && \forall\ s\in S,\ v\in V,\ t\in T && (5)\\
& 0 \le \hat{d}_v^t \le \hat{C}_v && \forall\ v\in V && (6)\\
& 0 \le d_s^t \le C_s && \forall\ s\in S && (7)\\
& 0 \le x_s^{+,t} \le f_s^{+,t}, 0 \le x_s^{-,t} \le f_s^{-,t} && \forall\ s\in S,\ t\in T && (8)\\
& 0 \le r_{s,v}^{+,t}, r_{s,v}^{-,t} \le \hat{C}_v && \forall\ s\in S,\ v\in V,\ t\in T && (9)\\
& z_{s,v}^t \in \{0,1\} && \forall\ s\in S,\ v\in V,\ t\in T. && (10)
\end{aligned}
$$

The above model, denoted as **D**ynamic **R**ebalancing **O**ptimization for **B**SS minimizing **L**ost **D**emand (DROB-LD), derives rebalancing strategies for the entire planning horizon, i.e., it decides how many bikes each vehicle should pick up or drop off at which station. The model can be easily implemented in different reoptimization modes (static, rolling, and folding planning) with alterable length of planning horizons and duration of time-periods, depending on the requirements of the decision-maker.

3.2. Rebalancing models based on inventory interval and target inventory

Even though the above used objective minimizing the lost demand is quite popular in the literature, its performance is sensitive to the accuracy of the expected rentals $f_s^{+,t}$ and returns $f_s^{-,t}$. Rather than minimizing the deviation from such a point estimate, we propose to minimize the deviation from either the inventory interval or the target inventory. This approach provides a buffer for the station inventories, allowing them to maintain reasonable inventories even when the trip prediction is less accurate, and to be better prepared for fluctuations of the stochastic demand.

To this end, we propose two multi-period models with novel objective functions: **D**ynamic **R**ebalancing **O**ptimization for **B**SS based on **T**arget Inventories (DROB-T) and **D**ynamic **R**ebalancing **O**ptimization for **B**SS based on **I**nventory Intervals (DROB-I). The parameters and variables used in both models are depicted in Table 3.

Table 3: Parameters and variables that define inventory intervals and target inventories

Input	Definition
ℓ_s^t	The target inventory of station $s \in S$ at period $t \in T$
$\underline{\ell}_s^t$	The lower bound for the inventory interval of station $s \in S$ at period $t \in T$
$\overline{\ell}_s^t$	The upper bound for inventory interval of station $s \in S$ at period $t \in T$
Variables	Definition
$e_s^{+,t}$	The number of bikes above the upper bound at station $s \in S$ in period $t \in T$
$e_s^{-,t}$	The number of bikes below the lower bound at station $s \in S$ in period $t \in T$

The objective function of DROB-T (11) aims at minimizing the total deviations between station inventories and target values, thus yielding the following formulation:

$$\begin{aligned} \min \quad & \sum_{s \in S} \sum_{t \in T} |\ell_s^t - d_s^t| && (11) \\ \text{s.t.} \quad & (2) - (10). \end{aligned}$$

DROB-I is designed to keep the station inventories as much as possible within the computed intervals. To this end, DROB-I is formulated as the objective function (12), along with constraints (2)-(10) and (13)-(15) as defined below:

$$\begin{aligned} \min \quad & \sum_{s \in S} \sum_{t \in T} e_s^{-,t} + e_s^{+,t} && && (12) \\ \text{s.t.} \quad & \underline{\ell}_s^t - e_s^{-,t} \le d_s^t && \forall\ s \in S,\ t \in T && (13) \\ & d_s^t \le \overline{\ell}_s^t + e_s^{+,t} && \forall\ s \in S,\ t \in T && (14) \\ & e_s^{-,t} \ge 0, e_s^{+,t} \ge 0 && \forall\ s \in S,\ t \in T && (15) \\ & (2) - (10). \end{aligned}$$

Figure 2 exemplifies the inventory of a station s, as well as the lower bound and upper bound of the inventory interval, and its target inventory. For DROB-I, the excess of inventory at each time-period ($e_s^{+,t}$ and $e_s^{-,t}$, respectively), with respect to the inventory interval $[\underline{\ell}_s^t, \overline{\ell}_s^t]$, is computed by constraints (13) and (14). For DROB-T, Figure 2 also illustrates the deviation $|\ell_s^t - d_s^t|$ from the current target value. By minimizing these deviations, DROB-I and DROB-T aim at providing safety buffers to the station inventory and are therefore more likely of being capable to deal with

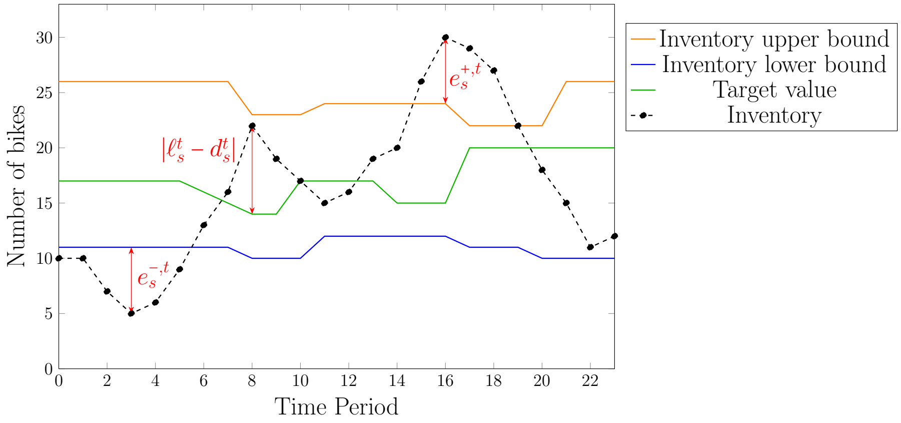

Figure 2: Example of deviations from inventory intervals and target inventory

stochastic demand fluctuations. We next present two toy examples to provide an intuition of the potential benefits of the two new objective functions.

Table 4: Comparative analysis of three objective functions for rebalancing operations

Objective Function	Example 1		Example 2	
	Dropped off Bikes	Expected Lost Demand*	Dropped off Bikes	Expected Lost Demand*
DROB-LD (1)	1	0.33	0	0.67
DROB-T (11)	8	0	5	0
DROB-I (12)	2	0	2	0

* The expected lost demand is calculated considering all possible chronological sequences of rentals and returns derived from historical trip data. For simplicity, we assume that the probability of a rental occurring before a return equals to that of a return happening before a rental.

Example 1. Consider an empty station with 10 docks. For a given time-period at a given day, historical rentals follow a uniform distribution ranging from 0 to 2, while no returns have been observed. A predictive model is used to predict the expected demand and target value and inventory interval are computed as to ensure a sufficient service level (see Section 4.1.2 and Appendix A for details). As a result, an estimation of 1 rental and no return is obtained, directly used in model DROB-LD. For DROB-T, the computed target value is 8, while for DROB-I, the computed inventory interval is [2, 10]. Table 4 summarizes the number of bikes dropped off at that station according to each of the three models. Furthermore, the table reports the expected lost demand if rental demand is uniformly distributed between 0 to 2. Here, DROB-T and DROB-I drop off at least 2 bikes, accounting for the potential demand of 2 rentals. In contrast, DROB-LD drops off only 1 bike, and therefore lacks 1 bike when the rental demand is 2.

Example 2. Consider that the same empty station, for another time-period and given day, has a uniform distribution between 0 and 2 for both rentals and returns. Both the estimated rental and return are therefore 1. The computed target value is 5, whereas the inventory interval is [2, 8]. In this case, DROB-T and DROB-I still drop off at least 2 bikes and therefore do not induce any unmet rental demand. In contrast, DROB-LD implicitly assumes that returns cancel rentals, and thus does not drop off any bikes, which may result in unmet demand when the rental demand is 1 or higher.

4. Experiments and Results

We now employ computational experiments to explore the benefits of the proposed models. Section 4.1 introduces the synthetic data and reports on the corresponding empirical results. A case study on real-world data is then presented in Section 4.2.

Computational environment. All optimization models are solved using IBM ILOG CPLEX v20.1.0.0 on 2.70 GHz Intel Xeon Gold 6258R machines with 8 cores. Optimization terminates once the MIP gap reaches 0.01% or the time limit of 24 hours is reached.

4.1. Experiments on synthetic data

We here focus on experiments carried out synthetic problem instances. To this end, Section 4.1.1 first introduces the instance generator that generates weather-dependent trip data. Section 4.1.2 then details the experimental set-up, including the machine learning model used to predict rental demand, the computation of inventory intervals and targets, and the simulator. This section also summarizes the computational results, comparing the performance of the various planning models. Finally, Section 4.1.3 then explores the performance of the planning models under the assumptions that predictions are less accurate.

4.1.1. Synthetic dataset

Even though we have access to real-world trip data, we synthetically generate instances for several reasons. First, the available real-world trip data lacks information on unobserved demand. Second, existing data may contain noise related to trip and station inventory data. Finally, rebalancing operations conducted by operators impact station inventory, but data on such operations is not openly available.

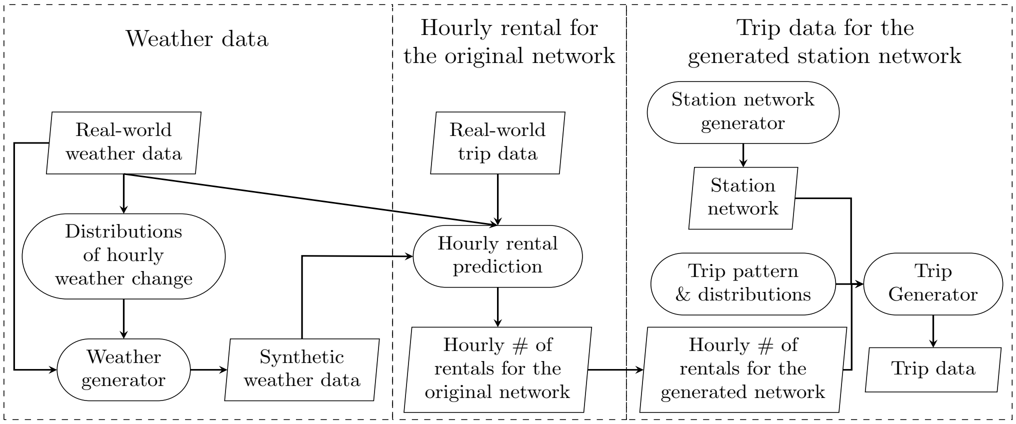
Weather data
Hourly rental for the original network
Trip data for the generated station network
Real-world weather data
Distributions of hourly weather change
Weather generator
Synthetic weather data
Real-world trip data
Hourly rental prediction
Hourly # of rentals for the original network
Station network generator
Station network
Trip pattern & distributions
Trip Generator
Hourly # of rentals for the generated network
Trip data

Figure 3: Process to generate weather data, station network and trip data

The general data generation process of the here employed instance generator is depicted in Figure 3. We extend the instance generator proposed by [36] (corresponding to the third box in Figure 3: "*Trip data for the generated station network*"), which is capable of generating diverse station networks and trip data, based on predefined *trip patterns & distributions* that align with those observed in real-world BSSs. Whereas the instance generator of [36] generates trip data under the same weather conditions (specifically, assuming high demand during summer months), we here explicitly acknowledge the strong correlation between weather conditions and trip demand. As such, we extend this instance generator as follows. (i) We introduce varying weather conditions into our generator (corresponding to the first box "*Weather data*" in Figure 3) by estimating statistical distributions that represent the hourly changes of weather conditions estimated on real-world weather data, and then sample new weather conditions from these distributions. (ii) We then compute the system-wide level of trip demand that is correlated to such weather conditions (corresponding to the second box "*Hourly rental for the original network*" in Figure 3), i.e., the hourly number of rentals for the entire network. Finally, individual trip demand is generated for the generated station network. Each of these components is next explained in detail.

Generation of weather data. Given that we ultimately aim at generating trip data that resembles periods of the demand peak season, we are interested in generating weather conditions for the months of June to August, which tend to have high user demand. We therefore procure *Real-world weather data*[1] from Montreal for June, July, and August from 2017 to 2020, resulting in a total of 368 days (including both weekdays and weekends). We select two features of utmost importance (see, e.g., [13; 17; 19; 31]): temperature and humidity. We analyze the temperature and humidity differences between consecutive hours throughout the day and divide the day into four distinct time segments such that temperature and humidity tend to remain relatively stable within each of which (0 am – 5 am, 6 am – 11 am, 12 pm – 5 pm, and 6 pm – 11 pm). The hourly differences are then used to estimate normal *Distributions of hourly weather change* for each time segment. The estimated distributions of temperature change for each time segment can be found in Appendix B.

Synthetic weather data has been generated for a total of 500 days as follows. For each hour of the original 368 days, we use the temperate and humidity that originally occurred at that day and add a change of temperature and humidity, respectively, sampled from the corresponding distributions. To obtain a total of 500 days, we repeat the process with the first 132 of the original 368 days. To ensure that the generated weather conditions are sufficiently realistic and avoid drastic fluctuations, we introduce constraints to keep the temperature and humidity within 5.5°C–36°C and 15%–99%, respectively. We denote the final set of the 500 generated days with synthetic weather data as the *Synthetic weather data*.

Generation of hourly rental demand. Using the temperature, humidity, hour, and weekday as features, we estimate the *Hourly rental prediction* model. This linear regression model is trained using the *Real-world weather data* and *Real-world trip data* (also see Section 4.2.1) and captures the correlation between time and weather conditions and the total demand level for the entire station network. The trained regression model is then used to estimate the total system-wide number of *Hourly # of rentals for the original network* that depends on the *Synthetic weather data* throughout each day. Given that this total number of rentals has been estimated on the original network from the *Real-world trip data* (here, the BIXI network with over 600 stations), this number is then scaled to the number of stations used in the here considered *Station network* (which has 60 stations). These hourly system-wide rental demands (*Hourly # of rentals for the generated network*) then serve as input to generate the detailed trip data.

Generation of station network and individual trip data. We generate two

[1] https://climate.weather.gc.ca

ground truth problem instances, denoted GT1 and GT2, each of which contains a *station network*, as well as hourly weather data and detailed trip information for 500 days. In both instances, the network contains 60 stations with different numbers of city center stations. The stations within city centers are equipped with 40 docks, while those outside city centers have 20 docks each.

Generated trips contain the origin station, the destination station, the departure time, and the arrival time. We consider four trip patterns of user behaviors with origins and destinations outside (O) and inside (I) city centers: (i) users who live outside city centers and work inside city centers typically use similar origin (outside city centers) and destination stations (inside city centers) during peak hours (OI trips); (ii) users who live and work outside city centers (OO trips); (iii) random non-work related trips occurring during the day (RD trips), and (iv) random non-work related trips occurring during the night (RN trips).

Table 5: Characteristics of the two considered ground truth instances

Instance		GT1	GT2
Network (60 stations)	# of city centers	1	2
	# of stations per city center	9	6
	City center capacity	26%	35%
Trip Pattern	OI	32%	32%
	OO	32%	32%
	RD	23%	23%
	RN	13%	13%

The characteristics of instance ground truths GT1 and GT2 are described in Table 5. Although the proportions of work-related (i.e., city center related) trips are identical in GT1 and GT2, the latter has more city center stations (12 stations in 2 city centers, as opposed to 9 stations in 1 city center). As such, work related trips in GT2 are distributed over a larger number of stations, which are therefore less stressed.

To sample individual trips for the considered *station network*, we assume that each trip type follows a particular temporal distribution, indicating the probabilistic time at which the rental occurs (as detailed in [36]). For each day, the *trip generator* then sequentially samples trips (i.e., origin-destination pairs and exact time stamps) from the *Trip pattern & distributions* (as defined by the ground truth) until the *hourly # of rentals for the generated network* is met, resulting in the final set of *Trip Data.*

4.1.2. Model performance based on regular trip predictions

Experimental set-up. The 500 generated days for GT1 and GT2 are separated into training set, validation set and test set as follows. The first 250 days are allocated to calibrate the gradient boosted tree introduced in [28], capable of predicting hourly station demand and trained on trips (time of rental and arrival/departure stations), weather conditions (temperature and humidity), and temporal data (day of the week, hour of the day, and a binary indicator for holidays). The subsequent 100 days are used for the validation and fine-tuning of the gradient boosted tree and inventory intervals. Details on the training of the gradient boosted tree can be found in Appendix A.

The remaining 150 days constitute the test set on which the optimization algorithms are executed. We consider a planning horizon from 7 a.m. to 3 p.m., discretized into 8 time-periods, each with a duration of one hour. For each of the 150 days, we assume to have access only to its corresponding weather conditions, but not to the exact trip (i.e., rental and return) demand. This is a reasonable assumption in practice, where one can assume to have access to a reasonably accurate weather prediction. Based on such weather conditions, the trained gradient boosted tree then predicts the hourly rental and return demand for each station ($f_s^{+,t}$ and $f_s^{-,t}$), used within model DROB-LD. The inventory intervals and target values, used within models DROB-I and DROB-T, are then computed based on the predicted rental demand (details can also be found in Appendix A). For each of the 150 days, the rebalancing planning solutions provided by the various models are then evaluated in the simulator (see Section 4.1.2) on the exact trip data. Note, again, that the optimization models only have access to demand predictions (based on weather data), whereas the simulator evaluates on the exact trip demand of the days in the test set.

In all experiments, 4 vehicles are available to rebalance the stations, each with a capacity for 40 bikes. The initial inventory of stations is obtained by solving an overnight rebalancing problem (equivalent to the one used in [36]).

Each of the optimization models can be executed in different reoptimization planning modes. In *static planning*, the optimization model is solved once for the entire planning horizon. The rebalancing strategies of the first 6 (out of 8) time-periods are then executed within the simulator to estimate the lost demand. The *rolling planning* has 3 optimization stages, each of which contemplates 4 time-periods. At each stage, the rebalancing decisions of the first 2 time-periods are executed within the simulator, as illustrated in Figure 1 (A). The *folding planning* uses all the remaining time-periods at each stage, as depicted in Figure 1 (B). The fine-grained discrete-event simulator from [36] here used employs a chronological first-arrive-first-serve rule, for both user rentals and returns, as well as rebalancing vehicles (i.e., pick-ups and drop-offs). Events are discretized events into 1-minute time-slots, which results in a particularly detailed and realistic simulation.

Computational Results. Table 6 illustrates the average lost demand and computing time (over the test set) for all the models and reoptimization modes (**S**tatic, **R**olling, and **F**olding). We report the computing times required to solve the optimization models as 'Opt. Time' (in minutes). The lost rental demand is computed as the relative gap between successful rentals and the original rental demand specified in the instances over the entire planning horizon, i.e., $\frac{\sum_{s,t}(f_s^{+,t}-\hat{x}_s^{+,t})}{\sum_{s,t} f_s^{+,t}}$, where $\hat{x}_s^{+,t}$ is the number of successful rentals in the simulator. 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 $\hat{x}_s^{+,t}$. We also present the relative difference ($\Delta(\%)$) of the rental, return, and total lost demand of DROB-I and DROB-T when compared to DROB-LD under the respective reoptimization mode.

Table 6 allows for the following observations:

1. **Comparison of proposed models.** From Table 6, models DROB-I and DROB-T generally outperform DROB-LD for both GT1 and GT2. For example, on GT1, DROB-I reduces the lost demand from 7.65% to 5.66% in rolling planning, while being solved within seconds. While under static planning, DROB-I and DROB-T tend to outperform DROB-LD, they consistently outperform DROB-LD by a higher rate under rolling planning (reducing total lost demand by 23.26%-34.51%)

2. **Comparison of different reoptimization modes.** The rolling and folding planning consistently result in lower lost demand than the static planning, likely due to the fact that they update the station inventories before reoptimizing at every reoptimization stage (see Figure 1). Updating the inventory narrows the gap between the estimated inventories in the optimization model and the observed inventories during the simulation, allowing the optimization models to make more informed decisions. For example, rolling and folding planning in DROB-I on GT1 reduce the lost demand from 7.43% to 5.66% and 5.74%, respectively, over the static planning. Note that updating weather forecast may have significat impact in the rolling planning, which will be discussed in Section 4.1.3. In terms of computing times, even though most models have been solved within 1-2 minutes, the folding planning requires much longer computing

Table 6: Results of dynamic rebalancing models for GT1 and GT2 under regular prediction

Instance	Model	Reopt.	Opt.Time	Lost Demand (%)					
		Mode	(min.)	Rental	Δ(%)	Return	Δ(%)	Total	Δ(%)
GT1	DROB-LD	S	0.19	13.14		2.74		8.31	
		R	0.02	11.94		2.76		7.65	
		F	0.21	11.54		2.87		7.48	
	DROB-I	S	0.18	10.68	▼ -18.72	3.79	▲ 38.32	7.43	▼ -10.59
		R	0.10	8.18	▼ -31.49	2.90	▲ 5.07	5.66	▼ -26.01
		F	0.39	8.39	▼ -27.30	2.84	▼ -1.05	5.74	▼ -23.26
	DROB-T	S	1.53	9.98	▼ -24.05	3.97	▲ 44.89	7.14	▼ -14.08
		R	0.15	8.20	▼ -31.32	1.52	▼ -44.93	5.01	▼ -34.51
		F	1.79	8.21	▼ -28.86	1.61	▼ -43.90	5.06	▼ -32.35
GT2	DROB-LD	S	0.41	8.27		1.68		5.13	
		R	0.06	8.16		1.67		5.06	
		F	0.44	8.11		1.62		5.01	
	DROB-I	S	0.04	7.76	▼ -6.17	2.51	▲ 49.40	5.24	▲ 2.14
		R	25.24	6.31	▼ -22.67	1.35	▼ -19.16	3.92	▼ -22.53
		F	229.15	5.87	▼ -27.62	1.12	▼ -30.86	3.57	▼ -28.74
	DROB-T	S	0.13	8.79	▲ 6.29	0.63	▼ -62.50	4.91	▼ -4.29
		R	0.28	6.96	▼ -14.71	0.58	▼ -65.27	3.89	▼ -23.12
		F	0.32	6.97	▼ -14.06	0.57	▼ -64.81	3.90	▼ -22.16

times for GT2.

3. **Comparison between GT1 and GT2.** While the general conclusions and tendencies are the same for GT1 and GT2, for GT1, models present more unmet demand than for GT2. Indeed, GT2 has more stations located in city centers, a region with high work-related demand, resulting in more evenly distributed trip patterns for these stations. Moreover, the larger number of city center stations translates into greater availability of docks in the network, as stations located in this area contain twice the number of docks than stations located in other regions of the city.

Results based on perfect trip prediction. We further carry out experiments with perfect trip information for each day of the test set, i.e., the model optimizes on the exact rental and return demand on which its rebalancing policy is later evaluated

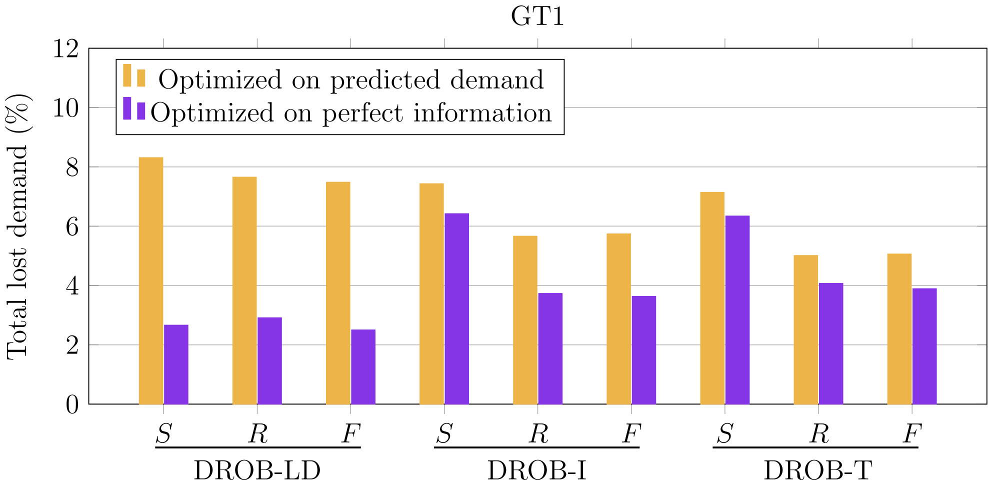

Figure 4: Results of total lost demand for GT1 using regular prediction and ground truth

within the simulator. These experiments establish an empirical performance bound and provide insights into the efficiency of the optimization models and reoptimization modes. Figure 4 presents the total lost demand obtained using the predicted trip demand (i.e., the same as used for Table 6) and the perfect information for GT1. Unsurprisingly, if perfect information was available, DROB-LD would consistently outperform DROB-I and DROB-T, given that an inventory safety buffer would be unnecessary. However, in reality, demand is stochastic. In this case, DROB-I and DROB-T can provide more robust station inventories, allowing them to deal with the stochastic trip demand. Interestingly, DROB-I and DROB-T still benefit from inventory updates (rolling and folding planning) under perfect information, as opposed to DROB-LD. Indeed, the former two models rely on the current inventory levels to update their objective functions, while the objective of DROB-LD remains unchanged, even when station inventories change. As a result, reoptimization for DROB-LD is not beneficial. Finally, the results also enable us to derive insights into the empirical bounds on the potential gains achieved through the utilization of a more accurate predictive model. While the gains are substantial (~5% of lost demand) for DROB-LD, they are much smaller (~1-2%) for DROB-I and DROB-T. While using a more accurate predictions may obviously lead to reduced unmet demand, we will next investigate how those models perform when predictions are less accurate.

4.1.3. Model performance based on noisy prediction

The optimization models used in our previous experiments have taken as input demand predictions and interval predictions that have assumed a perfect weather forecast. In practice, weather forecasts for the next 2 to 8 hours can be prone to inaccuracies. In a similar vein, having access to a predictive model with sufficiently high accuracy may not always be possible. We will now investigate the performance of the various models under the assumption that demand and interval predictions are less accurate. To this end, we deliberately introduce noise into the performed trip predictions. Since accurately predicting demand becomes increasingly challenging as we project further into the future, we introduce more noise to later time-periods.

Noisy predictions. We consider two types of effects caused by noises over demand predictions: (i) overestimation, e.g., due to a forecast of overly favorable weather conditions and, therefore, expecting a higher number of trips than will actually occur; (ii) underestimation, e.g., due to a forecast of adverse weather conditions, therefore predicting a lower number of trips than will actually occur.

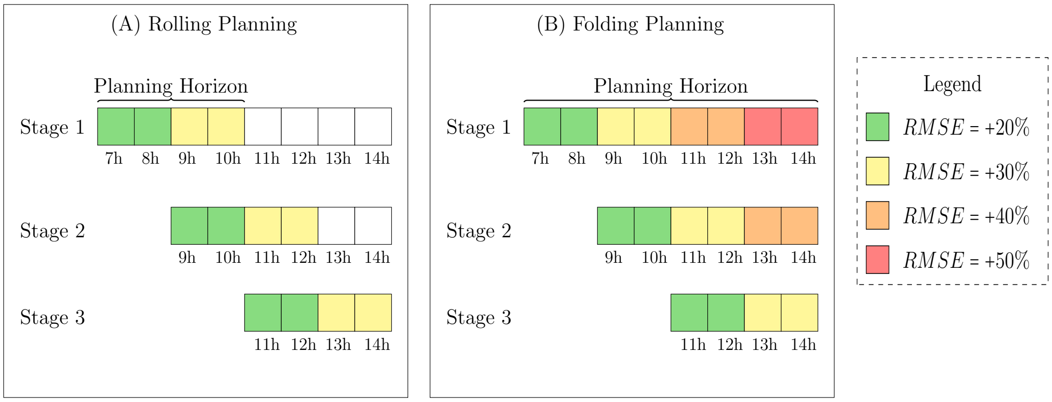

Figure 5: Noise added to the demand prediction over the planning horizon of the rolling and folding planning

To obtain less accurate predictions for the demand at station s at time t, we sample noise from a normal distribution. Its mean μ is determined by the original predicted number of rentals (or returns), while its standard deviation is strategically adjusted to achieve a predetermined increase in the Root Mean Squared Error (RMSE) for these noisy predictions in comparison to μ. This approach is applied throughout the planning horizon for each stage in the rolling and folding planning as illustrated in Figure 5. Values sampled above the mean are used to create overestimating predictions, whereas values below the mean are sampled to create underestimating predictions. Thus, an underestimating forecast consistently predicts lower demand and an overestimating forecast consistently predicts higher demand. Note that static planning operates under the same conditions as the first stage of folding planning.

Results. Tables 7 and 8 show the results for the three optimization models under underestimating and overestimating predictions, respectively. As previously in Table 6, Δ(%) indicates the relative difference of lost demand between DROB-I / DROB-T and baseline model DROB-LD.

Table 7: Results of dynamic rebalancing models for GT1 and GT2 under underestimating predictions

Instance	Model	Reopt.	Opt.Time	Lost Demand (%)					
		Mode	(min.)	Rental	Δ(%)	Return	Δ(%)	Total	Δ(%)
GT1	DROB-LD	S	0.02	13.97		4.56		9.62	
		R	0.02	11.90		3.54		7.99	
		F	0.04	11.57		3.56		7.81	
	DROB-I	S	11.45	10.31	▼ -26.20	4.37	▼ -4.17	7.51	▼ -21.93
		R	0.13	8.23	▼ -30.84	2.82	▼ -20.34	5.65	▼ -29.29
		F	21.13	7.73	▼ -33.19	1.68	▼ -52.81	4.84	▼ -38.03
	DROB-T	S	0.20	11.89	▼ -14.89	2.55	▼ -44.08	7.52	▼ -21.83
		R	0.30	9.98	▼ -16.13	1.11	▼ -68.64	5.78	▼ -27.66
		F	2.84	10.03	▼ -13.31	1.04	▼ -70.79	5.78	▼ -25.99
GT2	DROB-LD	S	0.01	11.66		2.69		7.46	
		R	0.01	10.85		1.36		6.39	
		F	0.02	10.44		1.38		6.17	
	DROB-I	S	0.23	8.6	▼ -26.24	2.48	▼ -7.81	5.68	▼ -23.86
		R	0.19	7.02	▼ -35.30	1.40	▲2.94	4.32	▼ -32.39
		F	1.94	6.99	▼ -33.05	1.39	▲0.72	4.30	▼ -30.31
	DROB-T	S	21.30	9.42	▼ -19.21	1.27	▼ -52.79	5.55	▼ -25.60
		R	0.79	7.42	▼ -31.61	0.56	▼ -58.82	4.13	▼ -35.37
		F	21.88	7.48	▼ -28.35	1.16	▼ -15.94	4.45	▼ -27.88

Based on Tables 7 and 8, we summarize our observations as follows:

1. **Comparison of models.** Although DROB-LD shows performance improvement in the case of overestimating predictions, DROB-I and DROB-T consis-

Table 8: Results of dynamic rebalancing models for GT1 and GT2 under overestimating predictions

Instance	Model	Reopt.	Opt.Time	Lost Demand (%)					
		Mode	(min.)	Rental	Δ(%)	Return	Δ(%)	Total	Δ(%)
GT1	DROB-LD	S	1.50	12.09		2.14		7.44	
		R	0.04	10.79		2.12		6.71	
		F	1.52	10.50		2.18		6.58	
	DROB-I	S	0.08	11.05	▼ -8.60	3.77	▲76.17	7.62	▲2.42
		R	0.06	8.75	▼ -18.91	1.62	▼ -23.58	5.35	▼ -20.27
		F	1.41	8.59	▼ -18.19	0.86	▼ -60.55	4.91	▼ -25.38
	DROB-T	S	0.06	11.05	▼ -8.60	3.94	▲84.11	7.71	▲3.63
		R	0.10	8.77	▼ -18.72	1.24	▼ -41.51	5.19	▼ -22.65
		F	0.26	8.75	▼ -16.67	1.21	▼ -44.50	5.16	▼ -21.58
GT2	DROB-LD	S	1.31	7.58		0.77		4.32	
		R	0.06	6.97		0.93		4.07	
		F	1.46	6.87		0.85		3.97	
	DROB-I	S	0.03	8.04	▲6.07	1.64	▲112.99	4.98	▲15.28
		R	0.03	7.00	▲0.43	0.60	▼ -35.48	3.93	▼ -3.44
		F	0.07	7.01	▲2.04	0.60	▼ -29.41	3.93	▼ -1.01
	DROB-T	S	0.08	8.63	▲13.85	1.56	▲102.6	5.27	▲21.99
		R	0.05	7.04	▲1.00	0.45	▼ -51.61	3.88	▼ -4.67
		F	0.16	7.06	▲2.77	0.46	▼ -45.88	3.89	▼ -2.02

tently demonstrate lower lost demand in most cases. Especially within rolling and folding planning, DROB-I and DROB-T outperform DROB-LD considerably. Their advantage is particularly pronounced when optimizing on underestimating predictions.

2. **Comparison of different reoptimization modes.** Generally, the improvement of lost demand when transitioning from static planning to folding and rolling planning is more significant under perturbed trip predictions than under noise-free predictions (see Table 6). This confirms the importance of such reoptimization planning modes when less accurate predictions are used.

3. **Comparison between predictions.** Underestimating trip predictions results in higher lost demand compared to noise-free predictions, since fewer rebalancing operations are triggered. In contrast, overestimating predictions may lead to less lost demand, especially notable for DROB-LD. This is explained by the

fact that overestimating predictions triggers more rebalancing operations in DROB-LD. We report the number of rebalancing operations (number of bikes picked up and dropped off) over the planning horizon in Figure 6. Indeed, overestimating predictions results in more rebalancing operations to meet the high demand. Overall, it appears that DROB-I and DROB-T are less sensitive to prediction noise than DROB-LD, given that they are designed to introduce a buffer into the optimized stations' inventories.

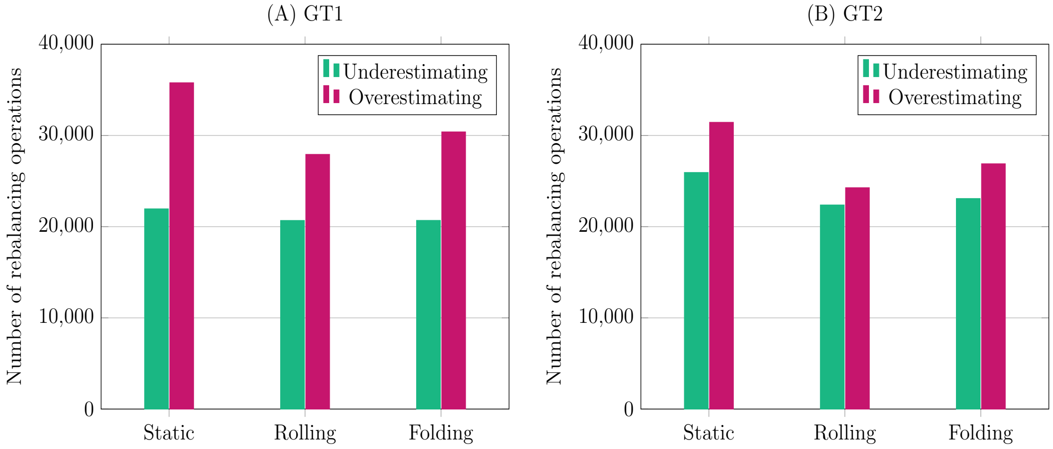

Figure 6: Number of rebalancing operations carried out in GT1 and GT2 for the DROB-LD

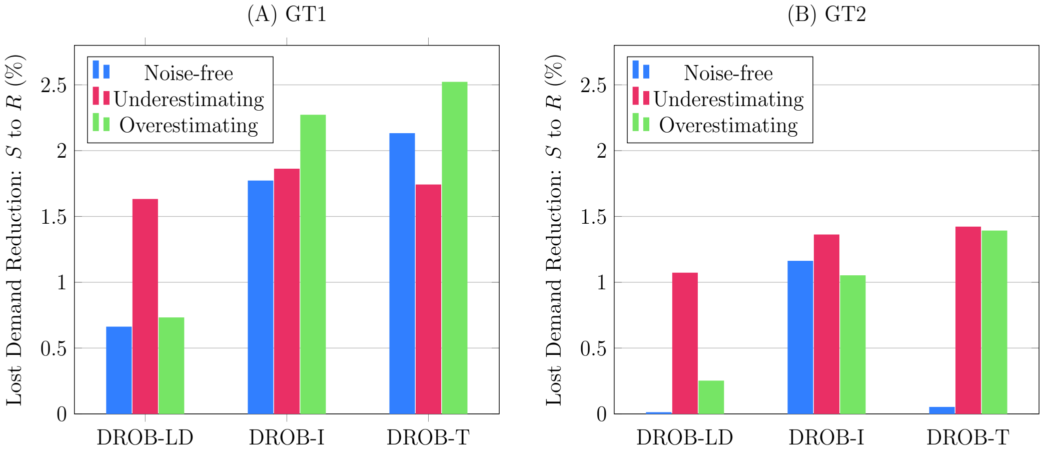

Figure 7: Improvement of the total lost demand from static to rolling planning

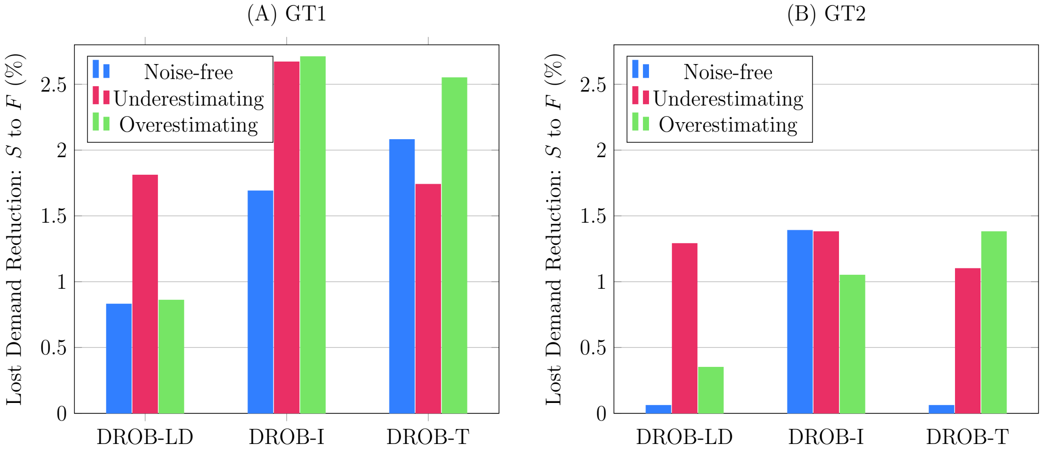

Figure 8: Improvement of the total lost demand from static to folding planning

Remarks. Overall, the results indicate that DROB-LD is much more susceptible to the accuracy of demand predictions than DROB-I and DROB-T as the directly considers such predictions in its objective function. By introducing a buffer to the stations inventories, the performance of DROB-I and DROB-T remains more stable among the different prediction approaches. To visualize the improvement of rolling/folding planning over static planning, we illustrate the difference in total lost demand between static and rolling planning in Figure 7, and between static and folding planning in Figure 8 for both GT1 and GT2. It is worth noting that the difference in total lost demand in these figures consistently shows positive values, meaning that in all experiments, rolling and folding planning consistently outperform static planning. These improvements are even more significant in the experiments with underestimating and overestimating predictions. This can be attributed to the fact that, in addition to updating the station inventory, a more accurate trip prediction is updated before re-optimization in each stage (see Figure 5).

4.2. Experiments on real-world data

In this section, we describe the real-world dataset used to validate the effectiveness of model DROB-I, which, on synthetic data, has demonstrated consistently low lost rentals while maintaining reasonable computing times. We first describe the real-world dataset in Section 4.2.1. We then describe the results in Section 4.2.2. The experimental set-up and planning horizon here considered are the same as in the experiments on synthetic data.

4.2.1. Real-world dataset

The real-world data consists of weather, temporal, station, and trip data. Weather and temporal data are directly provided by the official website of the Government of Canada, including temperature and humidity. The temporal data contains the date, hour (0h -23h), year (2019), and weekday (Monday to Friday). The trip and station data are provided by BIXI[2]. The trip data contain the origin station, start time, destination station, and arrival time of each trip, while the station data contain the location and station capacity (i.e., the number of docks).

We only focus on trips during weekdays from May to September 2019. Selecting trips before 2020 ensures that analyzed trip patterns are not affected by the COVID-19 pandemic. Weekdays are chosen due to their typically consistent work-related trip patterns. The first 21 days of each month, excluding the weekends, constitute the training dataset. The remaining days of May are used for validation and the remaining days from June to September are assigned to the test dataset. The initial inventory for stations at 7 a.m. is also collected from BIXI dataset and serves as input for the optimization models.

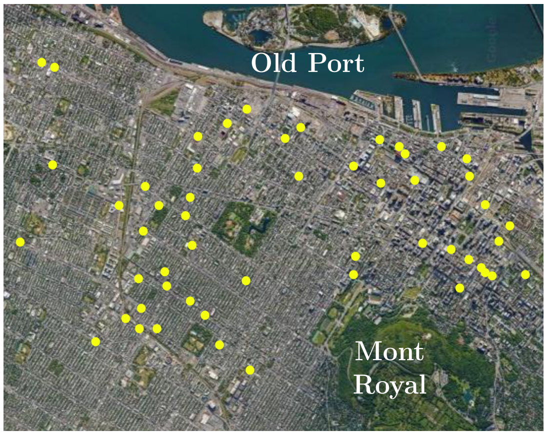

Figure 9: Cluster of BIXI stations located in Montreal (Canada)

For BIXI's station network, we exclude stations that have been relocated more than 1 km from their original locations by the operator during specific events, constructions, or holidays. As a result, 606 stations out of originally 620 remain in our experiments. This network is too large to be directly solved by general-purpose solvers. As a remedy, literature often divides the network into smaller clusters. Re-

[2] https://bixi.com/en/open-data-2/

balancing is then performed within a specific cluster or between different clusters (see, e.g., [6; 15; 20; 27; 30; 38]).

We follow the approach of [36] to cluster the stations according to their trip behaviour using k-means. We then select a cluster around the downtown and plateau areas in which the total number of rentals is approximately the same as the total number of returns. This cluster has 53 stations, including several city center stations and therefore contains work-related trips. Given that the distances between stations inside the cluster are limited, vehicles have sufficient time available to relocate and rebalance bikes within each time-period. The stations in the selected cluster are visualized in Figure 9.

4.2.2. Results on Real-world data

Given that, on synthetic data, DROB-I within rolling planning outperformed DROB-T on lost rental under all the predictions and consistently had swift computing times, we here focus on comparing the performance of DROB-I and DROB-LD. Experiments are carried out in a rolling planning, which aligns with practice and accommodates the need for swift runtime, while also allowing for real-time system updates.

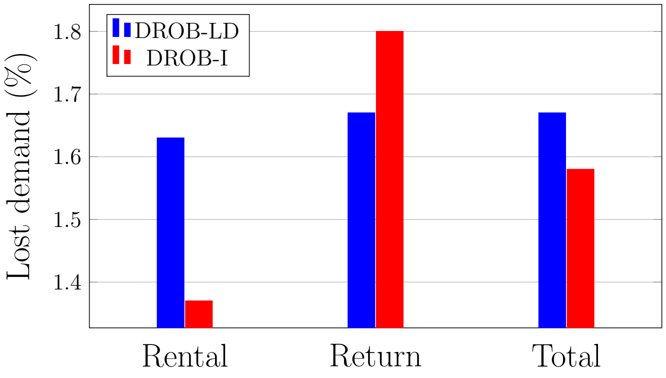

Figure 10: Lost demand of DROB-I and DROB-LD on a cluster from BIXI

The results are visualized in Figure 10. Detailed results for each day can be found in Appendix C. DROB-I performs better on both lost rental and total lost demand, while the lost return is higher than for DROB-LD. Note that lower lost rental demand (and therefore more succesful trips) also results in more return demand, which explains that DROB-I suffers from a slightly higher lost return. Such observations align with previous results on synthetic data. The results on real-world data confirm the benefits of model DROB-I, providing robustness to the station inventories and reducing the total lost demand. In terms of computing time, both models can be solved to optimality within 1 minute.

In contrast to the results on synthetic data, the improvement provided by DROB-I is not impressive. Such smaller improvement may be explained by the fact that the here-considered real-world data only contains successful trips. We also carried out static planning for the DROB-LD and DROB-I. However, under static planning, DROB-LD runs out of memory for many cases, given that the model contains more time-periods. The total lost demand of DROB-I under static planning has been 1.98%, shortly higher than the one under rolling planning (1.58%), highlighting once again the benefits of the rolling planning for DROB-I.

5. Conclusions

In this work, we have proposed two objective functions for multi-period rebalancing models for Bike-sharing systems, DROB-I and DROB-T, incorporating inventory intervals and target inventories. The resulting models provide an alternative to classical models minimizing unmet demand and are particularly suitable for BSS operators that use (often manually computed) inventory intervals and targets to guide their rebalancing process.

Our work evaluates the entire pipeline required for an automatized and data-driven rebalancing process. Instead of relying on manual input, we estimate rental and return demand for each hour and station in a data-driven fashion, using a machine learning model that has been shown to provide reasonably accurate results based on historical data related to time, weather and user trips. Inventory intervals and targets are then derived such that they maximize the desired service-level.

Our empirical analysis explore the capability of the proposed planning solutions to meet customer demands under three key characteristics of the planning process: the used optimization model, the employed reoptimization mode, and the impact of highly accurate (or inaccurate) demand predictions. The obtained planning solutions are then evaluated within a fine-grained minute-by-minute discrete-event simulator. A series of experiments on synthetic data allows for three key conclusions: First, our proposed models exhibit remarkable robustness compared to DROB-LD, the classical model minimizing unmet demand. DROB-T leads to a reduction in lost demand of up to 34%, while DROB-I decreases lost demand by up to 28%. Second, such robustness is also observed when demand predictions are less accurate, as these models introduce a conservative buffer into the station inventories, capable of better dealing with stochastic demand fluctuations. Third, there is a pronounced benefit in reoptimizing the rebalancing decisions throughout the planning as opposed to executing the optimization model only once and implementing a static planning solution for the entire planning horizon. Allowing for updated system information, the improvement via a rolling or folding planning has been found to be consistently in the order of 15-20% as opposed to static planning for DROB-LD. For DROB-I and DROB-T, the improvement tends to be higher than 30%, clearly indicating the benefits of such additional reoptimization effort. Finally, the benefits of our proposed models observed on synthetic data are also verified on a case study with real-world data from a BIXI Montreal.

Our approach may be highly attractive to system operators, not only due to their superior performance, but also due to their fit within the existing decision-making processes, as inventory intervals and targets are often used concepts in practice. In addition, we hope that the here proposed weather generator inspires future research to evaluate planning approaches in a more complex and realistic manner. Given the benefits of the here proposed models, we believe that the development of tailored solution methods (such as mathematical decomposition methods) may be promising research directions, highly useful for both academia and practitioners to approach rebalancing in large-scale station networks.

Acknowledgements

The authors are thankful to BIXI Montreal for providing real-world trip data and for their support throughout several discussions. We also thank the Digital Research Alliance of Canada for providing the computational resources to carry out the numerical experiments. The Natural Sciences and Engineering Research Council (NSERC) of Canada has supported the third and fourth author under grants 2023-04466 and 2017-05224, respectively. The fourth author was also supported by the Fonds de Recherche du Québec - Nature et technologie (FRQNT).

References

[1] H. Akova, S. Hulagu, and H. B. Celikoglu, “Static bike repositioning problem with heterogeneous distribution characteristics in bike sharing systems,” *Transportation Research Procedia*, vol. 62, pp. 205–212, 2022.

[2] R. Alvarez-Valdes, J. M. Belenguer, E. Benavent, J. D. Bermudez, F. Muñoz, E. Vercher, and F. Verdejo, “Optimizing the level of service quality of a bike-sharing system,” *Omega*, vol. 62, pp. 163–175, 2016.

[3] P. Borgnat, P. Abry, P. Flandrin, C. Robardet, J.-B. Rouquier, and E. Fleury, "Shared bicycles in a city: A signal processing and data analysis perspective," *Advances in Complex Systems*, vol. 14, no. 03, pp. 415–438, 2011.

[4] N. Boufidis, A. Nikiforiadis, K. Chrysostomou, and G. Aifadopoulou, "Development of a station-level demand prediction and visualization tool to support bike-sharing systems' operators," *Transportation Research Procedia*, vol. 47, pp. 51–58, 2020.

[5] J. Brinkmann, M. W. Ulmer, and D. C. Mattfeld, "The multi-vehicle stochastic-dynamic inventory routing problem for bike sharing systems," *Business Research*, vol. 13, no. 1, pp. 69–92, 2020.

[6] G. C. Calafiore, C. Bongiorno, and A. Rizzo, "A robust mpc approach for the rebalancing of mobility on demand systems," *Control Engineering Practice*, vol. 90, pp. 169–181, 2019.

[7] D. Chemla, F. Meunier, and R. W. Calvo, "Bike sharing systems: Solving the static rebalancing problem," *Discrete Optimization*, vol. 10, no. 2, pp. 120–146, 2013.

[8] Q. Chen, C. Fu, N. Zhu, S. Ma, and Q.-C. He, "A target-based optimization model for bike-sharing systems: From the perspective of service efficiency and equity," *Transportation Research Part B: Methodological*, vol. 167, pp. 235–260, 2023.

[9] Y. Chumin, O. O'Brien, P. DeMaio, R. Rabello, S. Chou, and T. Benicchio, "The Meddin bike-sharing world map report," https://bikesharingworldmap.com/reports/bswm_mid2021report.pdf, 2021, accessed: 2023-08-04.

[10] C. Contardo, C. Morency, and L.-M. Rousseau, *Balancing a dynamic public bike-sharing system.* Cirrelt Montreal, Canada, 2012, vol. 4.

[11] S. Datner, T. Raviv, M. Tzur, and D. Chemla, "Setting inventory levels in a bike sharing network," *Transportation Science*, vol. 53, no. 1, pp. 62–76, 2019.

[12] W. El-Assi, M. Salah Mahmoud, and K. Nurul Habib, "Effects of built environment and weather on bike sharing demand: a station level analysis of commercial bike sharing in toronto," *Transportation*, vol. 44, no. 3, pp. 589–613, 2017.

[13] E. Eren and V. E. Uz, "A review on bike-sharing: The factors affecting bike-sharing demand," *Sustainable cities and society*, vol. 54, p. 101882, 2020.

[14] S. Feng, H. Chen, C. Du, J. Li, and N. Jing, "A hierarchical demand prediction method with station clustering for bike sharing system," in *2018 IEEE Third International Conference on Data Science in Cyberspace (DSC)*. IEEE, 2018, pp. 829–836.

[15] I. A. Forma, T. Raviv, and M. Tzur, "A 3-step math heuristic for the static repositioning problem in bike-sharing systems," *Transportation research part B: methodological*, vol. 71, pp. 230–247, 2015.

[16] J. E. Froehlich, J. Neumann, and N. Oliver, "Sensing and predicting the pulse of the city through shared bicycling," in *Twenty-first international joint conference on artificial intelligence*, 2009, pp. 1420–1426.

[17] C. Gallop, C. Tse, and J. Zhao, "A seasonal autoregressive model of vancouver bicycle traffic using weather variables," *i-Manager's Journal on Civil Engineering*, vol. 1, no. 4, p. 9, 2011.

[18] D. Gammelli, Y. Wang, D. Prak, F. Rodrigues, S. Minner, and F. C. Pereira, "Predictive and prescriptive performance of bike-sharing demand forecasts for inventory management," *Transportation Research Part C: Emerging Technologies*, vol. 138, p. 103571, 2022.

[19] K. Gebhart and R. B. Noland, "The impact of weather conditions on bikeshare trips in Washington, DC," *Transportation*, vol. 41, no. 6, pp. 1205–1225, 2014.

[20] S. Ghosh, P. Varakantham, Y. Adulyasak, and P. Jaillet, "Dynamic redeployment to counter congestion or starvation in vehicle sharing systems," in *Twenty-Fifth International Conference on Automated Planning and Scheduling*, 2015.

[21] S. Ghosh, M. Trick, and P. Varakantham, "Robust repositioning to counter unpredictable demand in bike sharing systems," in *Proceedings of the 25th International Joint Conference on Artificial Intelligence IJCAI 2016*. AAAI Press, 2016, pp. 3096–3102.

[22] S. Ghosh, P. Varakantham, Y. Adulyasak, and P. Jaillet, "Dynamic repositioning to reduce lost demand in bike sharing systems," *Journal of Artificial Intelligence Research*, vol. 58, pp. 387–430, 2017.

[23] S. Ghosh, J. Y. Koh, and P. 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*, 2019, pp. 5864–5870.

[24] M. D. Gleditsch, K. Hagen, H. Andersson, S. J. Bakker, and K. Fagerholt, "A column generation heuristic for the dynamic bicycle rebalancing problem," *European Journal of Operational Research*, 2022, in press.

[25] R. Hu, Z. Zhang, X. Ma, and Y. Jin, "Dynamic rebalancing optimization for bike-sharing system using priority-based moea/d algorithm," *IEEE Access*, vol. 9, pp. 27 067–27 084, 2021.

[26] J. Huang, H. Sun, H. Li, L. Huang, A. Li, and X. Wang, "Central station-based demand prediction for determining target inventory in a bike-sharing system," *The Computer Journal*, vol. 65, no. 3, pp. 573–588, 08 2020.

[27] J. Huang, Q. Tan, H. Li, A. Li, and L. Huang, "Monte carlo tree search for dynamic bike repositioning in bike-sharing systems," *Applied Intelligence*, pp. 1–16, 2022.

[28] P. Hulot, D. Aloise, and S. D. Jena, "Towards station-level demand prediction for effective rebalancing in bike-sharing systems," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 378–386.

[29] G. Héctor, L. Rafael, and A. Ramirez-Nafarrate, "A simulation-optimization study of the inventory of a bike-sharing system: The case of mexico city ecobici's system," *Case Studies on Transport Policy*, vol. 9, no. 3, pp. 1059–1072, 2021.

[30] Y. Jin, C. Ruiz, and H. Liao, "A simulation framework for optimizing bike rebalancing and maintenance in large-scale bike-sharing systems," *Simulation Modelling Practice and Theory*, vol. 115, p. 102422, 2022.

[31] K. Kim, "Investigation on the effects of weather and calendar events on bike-sharing according to the trip patterns of bike rentals of stations," *Journal of transport geography*, vol. 66, pp. 309–320, 2018.

[32] C. Kloimüllner, P. Papazek, B. Hu, and G. R. Raidl, "Balancing bicycle sharing systems: an approach for the dynamic case," in *European Conference on Evolutionary Computation in Combinatorial Optimization.* Springer, 2014, pp. 73–84.

[33] B. Legros, “Dynamic repositioning strategy in a bike-sharing system; how to prioritize and how to rebalance a bike station,” *European Journal of Operational Research*, vol. 272, no. 2, pp. 740–753, 2019.

[34] G. Li, N. Cao, P. Zhu, Y. Zhang, Y. Zhang, L. Li, Q. Li, and Y. Zhang, “Towards smart transportation system: A case study on the rebalancing problem of bike sharing system based on reinforcement learning,” *Journal of Organizational and End User Computing (JOEUC)*, vol. 33, no. 3, pp. 35–49, 2021.

[35] Y. Li, Y. Zheng, and Q. Yang, “Dynamic bike reposition: A spatio-temporal reinforcement learning approach,” in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 1724–1733.

[36] J. Liang, S. D. Jena, and A. Lodi, “Dynamic rebalancing optimization for bike-sharing systems: A modeling framework and empirical comparison,” GERAD, HEC Montréal, Canada, Les Cahiers du GERAD G–2023–47, 2023.

[37] S. Lin, F. Y. Chen, Y. Li, and Z.-J. M. Shen, “Dynamic inventory control with covariates: Risk constraints, regularization, and folding-horizon plan,” *Yanzhi and Shen, Zuo-Jun Max, Dynamic Inventory Control with Covariates: Risk Constraints, Regularization, and Folding-horizon Plan (February 14, 2022)*, 2022.

[38] J. Liu, L. Sun, W. Chen, and H. 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*, 2016, pp. 1005–1014.

[39] M. Lowalekar, P. Varakantham, S. Ghosh, S. D. Jena, and P. Jaillet, “Online repositioning in bike sharing systems,” in *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 27, 2017, pp. 200–208.

[40] Á. Lozano, J. F. De Paz, G. Villarrubia Gonzalez, D. H. D. L. Iglesia, and J. Bajo, “Multi-agent system for demand prediction and trip visualization in bike sharing systems,” *Applied Sciences*, vol. 8, no. 1, p. 67, 2018.

[41] C.-C. Lu, “Robust multi-period fleet allocation models for bike-sharing systems,” *Networks and Spatial Economics*, vol. 16, no. 1, pp. 61–82, 2016.

[42] K. Mellou and P. Jaillet, "Dynamic resource redistribution and demand estimation: An application to bike sharing systems," *Available at SSRN 3336416*, 2019.

[43] O. O'Brien, P. DeMaio, R. Rabello, S. Chou, and T. Benicchio, "The meddin bike-sharing world map report," https://bikesharingworldmap.com/reports/bswm_mid2022report.pdf, 2022, accessed: 2023-08-04.

[44] A. Pal and Y. Zhang, "Free-floating bike sharing: Solving real-life large-scale static rebalancing problems," *Transportation Research Part C: Emerging Technologies*, vol. 80, pp. 92–116, 2017.

[45] Y. Pan, R. C. Zheng, J. Zhang, and X. Yao, "Predicting bike sharing demand using recurrent neural networks," *Procedia computer science*, vol. 147, pp. 562–566, 2019.

[46] M. Rainer-Harbach, P. Papazek, G. R. Raidl, B. Hu, and C. Kloimüllner, "Pilot, grasp, and vns approaches for the static balancing of bicycle sharing systems," *Journal of Global Optimization*, vol. 63, no. 3, pp. 597–629, 2015.

[47] T. Raviv and O. Kolka, "Optimal inventory management of a bike-sharing station," *IIE Transactions*, vol. 45, no. 10, pp. 1077–1093, 2013.

[48] T. Raviv, M. Tzur, and I. A. Forma, "Static repositioning in a bike-sharing system: models and solution approaches," *EURO Journal on Transportation and Logistics*, vol. 2, no. 3, pp. 187–229, 2013.

[49] V. Sathishkumar, P. Jangwoo, and C. Yongyun, "Using data mining techniques for bike sharing demand prediction in metropolitan city," *Computer Communications*, vol. 153, pp. 353–366, 2020.

[50] H. Sayarshad, S. Tavassoli, and F. Zhao, "A multi-periodic optimization formulation for bike planning and bike utilization," *Applied Mathematical Modelling*, vol. 36, no. 10, pp. 4944–4951, 2012.

[51] J. Schuijbroek, R. C. Hampshire, and W.-J. Van Hoeve, "Inventory rebalancing and vehicle routing in bike sharing systems," *European Journal of Operational Research*, vol. 257, no. 3, pp. 992–1004, 2017.

[52] Y. Seo, "A dynamic rebalancing strategy in public bicycle sharing systems based on real-time dynamic programming and reinforcement learning," Ph.D. dissertation, Doctoral dissertation. Seoul National University, South Korea, 2020.

[53] J. Shu, M. C. Chou, Q. Liu, C.-P. Teo, and I.-L. Wang, "Models for effective deployment and redistribution of bicycles within public bicycle-sharing systems," *Operations Research*, vol. 61, no. 6, pp. 1346–1359, 2013.

[54] C. Shui and W. Szeto, "Dynamic green bike repositioning problem–a hybrid rolling horizon artificial bee colony algorithm approach," *Transportation Research Part D: Transport and Environment*, vol. 60, pp. 119–136, 2018.

[55] Q. Tang, Z. Fu, D. Zhang, H. Guo, and M. Li, "Addressing the bike repositioning problem in bike sharing system: a two-stage stochastic programming model," *Scientific Programming*, vol. 2020, pp. 1–12, 2020.

[56] P. Vogel, *Service Network Design of Bike Sharing Systems.* Cham: Springer International Publishing, 2016, pp. 113–135.

[57] P. Vogel, T. Greiser, and D. C. Mattfeld, "Understanding bike-sharing systems using data mining: Exploring activity patterns," *Procedia-Social and Behavioral Sciences*, vol. 20, pp. 514–523, 2011.

[58] P. Vogel, B. A. Neumann Saavedra, and D. C. Mattfeld, "A hybrid metaheuristic to solve the resource allocation problem in bike sharing systems," in *Hybrid Metaheuristics: 9th International Workshop, HM 2014, Hamburg, Germany, June 11-13, 2014. Proceedings 9.* Springer, 2014, pp. 16–29.

[59] X. Wu, C. Lyu, Z. Wang, and Z. Liu, "Station-level hourly bike demand prediction for dynamic repositioning in bike sharing systems," in *Smart transportation systems 2019.* Springer, 2019, pp. 19–27.

[60] Y.-C. Yin, C.-S. Lee, and Y.-P. Wong, "Demand prediction of bicycle sharing systems," 2012, [Online].

[61] P.-S. You, "A two-phase heuristic approach to the bike repositioning problem," *Applied Mathematical Modelling*, vol. 73, pp. 651–667, 2019.

[62] K. R. Zamir, "Dynamic repositioning for bikesharing systems," Ph.D. dissertation, University of Maryland, College Park, 2020.

[63] D. Zhang, C. Yu, J. Desai, H. Lau, and S. Srivathsan, “A time-space network flow approach to dynamic repositioning in bicycle sharing systems,” *Transportation research part B: methodological*, vol. 103, pp. 188–207, 2017.

[64] J. Zhang, M. Meng, Y. D. Wong, P. Ieromonachou, and D. Z. Wang, “A data-driven dynamic repositioning model in bicycle-sharing systems,” *International Journal of Production Economics*, vol. 231, p. 107909, 2021.

[65] X. Zheng, M. Tang, Y. Liu, Z. Xian, and H. H. Zhuo, “Repositioning bikes with carrier vehicles and bike trailers in bike sharing systems,” *Applied Sciences*, vol. 11, no. 16, p. 7227, 2021.

Appendix A. Trip Prediction and Inventory Interval

The methodology for predicting trips and calculating inventory intervals and target inventories is adapted from the model presented by [28]. The rentals and returns ($f_s^{+,t}$ and $f_s^{-,t}$) are predicted on an hourly basis for each station. The model utilizes a Gradient Boosting Tree, which incorporates weather conditions (temperature and humidity) and temporal information (the day of the week, hour of the day, and holidays) as learning features. In this model, a Singular Value Decomposition (SVD) technique is applied to reduce the dimension of the trip data. This process results in faster predictions, enhancing tractability when dealing with an elevated number of stations. The SVD also elevates the accuracy of the model, indicated by a lower Root Mean Square Error (RMSE), as it effectively eliminates noise and outliers from the trip data. Figure A.1 illustrates the model's pipeline.

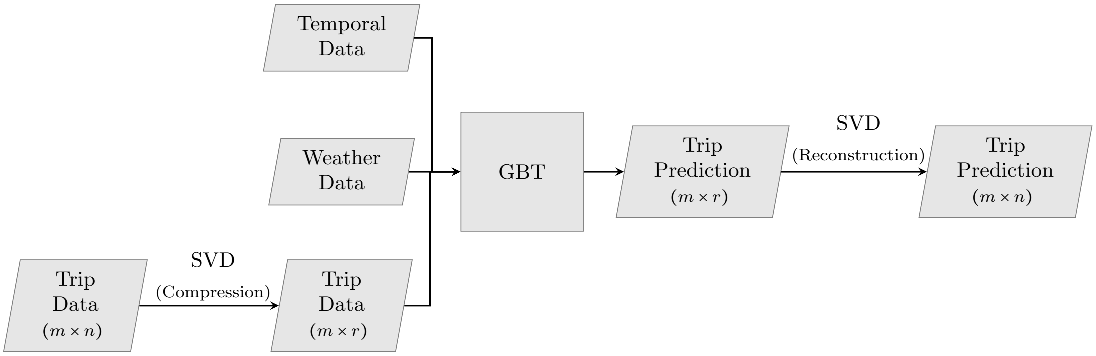

Figure A.1: Pipeline of the predictive model

Based on the predicted rental and return demand, the expected proportion of satisfied trips, known as *service level*, is computed for a given initial inventory. Assuming a station s with initial inventory i and capacity C_s, the rental and the return service levels for a time period $[t, t + \Delta]$ are computed as:

$$SL_s^{+,t}(i) = \frac{\int_t^{t+\Delta} f_s^{+,t}(1 - p_s^t(i,0))dt}{\int_t^{t+\Delta} f_s^{+,t}dt} \tag{A.1}$$

$$SL_s^{-,t}(i) = \frac{\int_t^{t+\Delta} f_s^{-,t}(1 - p_s^t(i,C_s))dt}{\int_t^{t+\Delta} f_s^{-,t}dt}, \tag{A.2}$$

where $f_s^{+,t}$ and $f_s^{-,t}$ represent the rental and return rates, respectively, for station s at time t. Further, $p_s^t(i,0)$, and $p_s^t(i,C_s)$ represent the probability that station s becomes empty and full, respectively, given an initial inventory i at time t.

The *overall service* level is computed as (A.3)

$$SL_s^t(i) = min\{SL_s^{+,t}(i), SL_s^{-,t}(i)\} \tag{A.3}$$

The minimum and maximum service levels, for a station s in time period $[t, t+\Delta]$, can then be computed depending on the initial inventory at time t as follows:

$$SL_s^{min,t} = min_{i\in\{0,\dots,C_s\}}(SL_s^t(i)) \tag{A.4}$$

$$SL_s^{max,t} = max_{i\in\{0,\dots,C_s\}}(SL_s^t(i)). \tag{A.5}$$

A threshold Ω is created to establish an acceptable service level for the time horizon $[t, t+\Delta]$:

$$\Omega_s^t = SL_s^{min,t} + \beta(SL_s^{max,t} - SL_s^{min,t}), \tag{A.6}$$

in which the hyperparameter β controls the proximity of the threshold Ω_s^t to either the minimum service level or the maximum service level. In practice, this hyperparameter influences the gap between the upper and the lower bound of the inventory interval.

The inventory interval for station s for time period $[t, t+\Delta]$ is then defined as

$$\mathcal{I}_s = \{i \in \{0, ..., C_s\} | \mathcal{L} \leq i \leq \mathcal{U}\}, \tag{A.7}$$

where $\mathcal{L} = \min\{i \in \{0, ..., C_s\} | SL_s^t(i) \geq \Omega_s^t\}$, and $\mathcal{U} = \max\{i \in \{0, ..., C_s\} | SL_s^t(i) \geq \Omega_s^t\}$. Finally, the target inventory for station s at time-period $[t, t+\Delta]$ is set to the initial inventory that results in the maximum service level (i.e., $SL_s^{max,t}$).

Appendix B. Weather Generator

The weather generator in Figure 3 utilizes normal distributions derived from the temperature and humidity differences between consecutive hours throughout the day, capturing the change in weather conditions over time.

Histograms of the temperature differences observed between two consecutive hours of 4 periods during the day (0 am – 5 am, 6 am – 11 am, 12 pm – 5 pm, and 6 pm – 11 pm) are depicted in Figure B.1. The overlaid red curves illustrate the normal distributions in which the parameters are computed using the Maximum Likelihood Estimator.

Figure B.1(A) and Figure B.1(C) display narrower distributions, suggesting less variability in temperature change, whereas Figure B.1(B) and Figure B.1(D) exhibit wider spreads, indicating greater fluctuation. The distributions for humidity differences are obtained using the same approach.

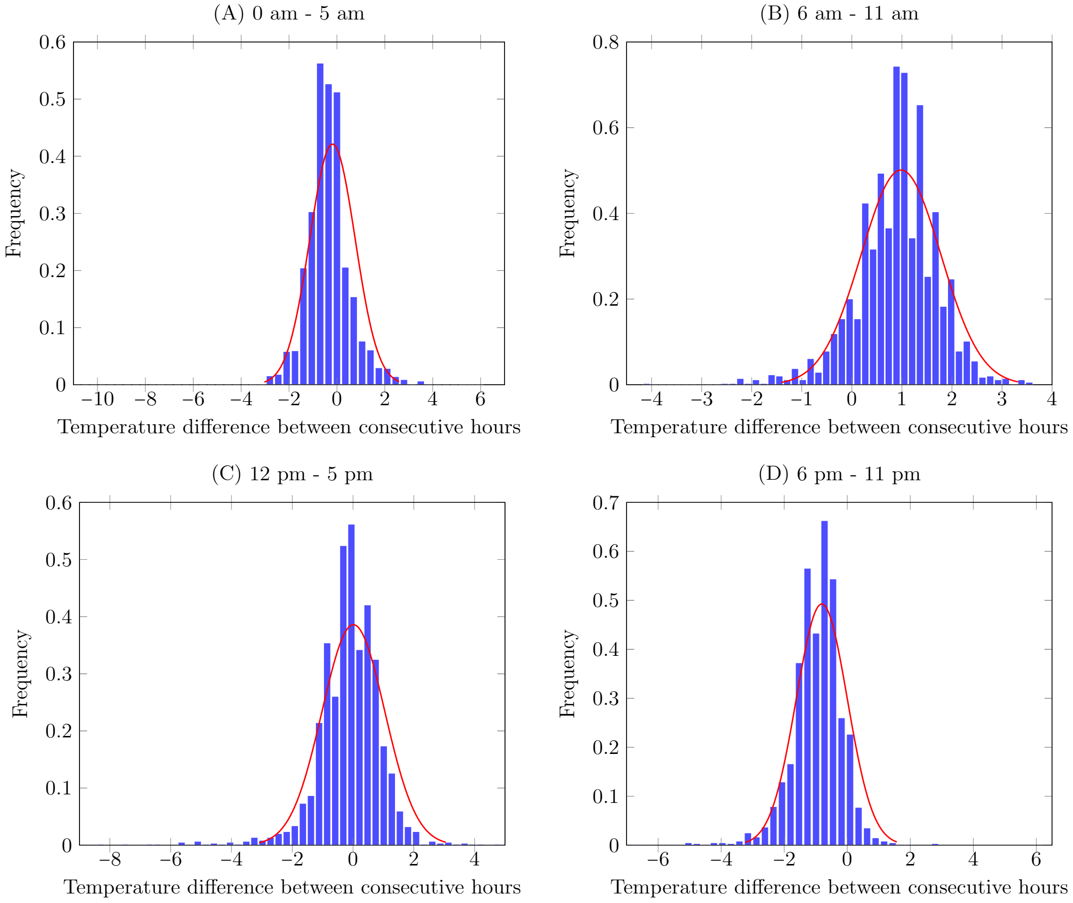

Figure B.1: Distributions of temperature changes between consecutive hours for four time-periods

Appendix C. Experimental Results on BIXI Cluster

The daily lost demand of models DROB-LD and DROB-I on the considered BIXI cluster are detailed in Table C.1.

Table C.1: Lost demand for DROB-LD and DROB-I on BIXI cluster

Day	Model	Opt.Time (min)	MIP Gap (%)	Lost demand (%)		
				Rental	Return	Total
Day 1	DROB-LD	0.00	0.00	0.85	7.52	4.40
	DROB-I	0.01	0.00	0.00	2.26	1.20
Day 2	DROB-LD	0.66	0.00	0.50	0.55	0.52
	DROB-I	0.41	0.00	0.50	0.55	0.52
Day 3	DROB-LD	5.41	0.00	0.73	0.98	0.85
	DROB-I	0.01	0.70	0.73	1.96	1.33
Day 4	DROB-LD	0.06	0.00	2.96	1.51	2.27
	DROB-I	0.01	0.00	2.09	1.51	1.81
Day 5	DROB-LD	0.02	0.00	1.74	2.33	2.01
	DROB-I	0.01	0.00	1.90	4.83	3.27
Day 6	DROB-LD	0.01	0.00	0.16	0.18	0.17
	DROB-I	0.01	0.00	0.00	1.41	0.66
Day 7	DROB-LD	0.04	0.00	0.35	2.92	1.56
	DROB-I	0.01	0.00	0.17	3.31	1.65
Day 8	DROB-LD	0.01	0.00	2.66	0.65	1.76
	DROB-I	0.01	0.00	2.66	2.39	2.54
Day 9	DROB-LD	0.09	0.00	1.75	0.80	1.30
	DROB-I	0.01	0.00	1.92	1.59	1.77
Day 10	DROB-LD	0.02	0.00	2.90	2.49	2.71
	DROB-I	0.01	0.00	2.90	2.49	2.71
Day 11	DROB-LD	0.01	0.00	1.71	0.73	1.23
	DROB-I	0.01	0.00	1.54	1.63	1.58
Day 12	DROB-LD	0.02	0.00	0.65	1.65	1.12
	DROB-I	0.01	0.00	1.13	0.74	0.95
Day 13	DROB-LD	0.13	0.00	3.58	1.42	2.52
	DROB-I	0.03	0.00	2.21	0.18	1.22
Day 14	DROB-LD	0.01	0.00	2.62	1.32	1.99
	DROB-I	0.01	0.00	2.62	1.81	2.23
Day 15	DROB-LD	0.01	0.00	0.85	2.21	1.50
	DROB-I	0.01	0.00	0.68	0.37	0.53
Day 16	DROB-LD	0.02	0.00	0.53	3.03	1.81
	DROB-I	0.02	0.00	0.00	3.03	1.55
Day 17	DROB-LD	0.11	0.00	2.56	1.12	1.84
	DROB-I	0.01	0.00	2.28	0.00	1.13
Day 18	DROB-LD	0.36	0.00	2.80	0.76	1.79
	DROB-I	0.03	0.00	2.24	3.82	3.02
Day 19	DROB-LD	0.01	0.00	2.34	0.20	1.29
	DROB-I	0.01	0.00	0.78	1.62	1.19
Day 20	DROB-LD	0.01	0.00	0.40	1.08	0.72
	DROB-I	0.01	0.00	1.00	0.43	0.72