EURO Journal on Transportation and Logistics 13 (2024) 100147

Contents lists available at ScienceDirect

EURO Journal on Transportation and Logistics

journal homepage: www.elsevier.com/locate/ejtl

Dynamic rebalancing for Bike-sharing systems under inventory interval and target predictions

Jiaqi Liang [a,c,d,e], Maria Clara Martins Silva [a,e], Daniel Aloise [a,d,e], Sanjay Dominik Jena [b,c,d,*]

[a] Polytechnique Montréal, 2500 Chemin de Polytechnique, H3T 1J4, Montréal, Canada
[b] School of Management, Université du Québec à Montréal, 315 Rue Sainte-Catherine Est, Montréal, H2X 3X2, Canada
[c] Centre interuniversitaire de recherche sur les reseaux d'entreprise, la logistique et le transport (CIRRELT), 2920, Chemin de la Tour, H3T 1J4, Montréal, Canada
[d] Canada Excellence Research Chair in Data-Science for Real-time Decision-Making, 2500 Chemin de Polytechnique, H3T 1J4, Montréal, Canada
[e] Groupe d'études et de recherche en analyse des décisions (GERAD), 2920, Chemin de la Tour, H3T 1N8, Montréal, Canada

ARTICLE INFO

Keywords:
Bike-sharing systems
Dynamic rebalancing
Inventory intervals
Target inventories
Reoptimization modes
Mixed-integer programming

ABSTRACT

Bike-sharing systems have become a popular transportation alternative. Unfortunately, station networks are often unbalanced, with some stations being empty, while others being congested. Given the complexity of the underlying planning problems to rebalance station inventories via trucks, many mathematical optimizations models have been proposed, mostly focusing on minimizing the unmet demand. This work explores the benefits of two alternative objectives, which minimize the deviation from an inventory interval and a target inventory, respectively. While the concepts of inventory intervals and targets better fit the planning practices of many system operators, they also naturally introduce a buffer into the station inventory, therefore better responding to stochastic demand fluctuations. We report on extensive computational experiments, evaluating the entire pipeline required for an automatized and data-driven rebalancing process: the use of synthetic and real-world data that relies on varying weather conditions, the prediction of demand and the computation of inventory intervals and targets, different reoptimization modes throughout the planning horizon, and an evaluation within a fine-grained simulator. Results allow for unanimous conclusions, indicating that the proposed approaches reduce unmet demand by up to 34% over classical models.

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 1900 BSSs in operation comprising almost 9 million bikes (O'Brien et al., 2022), offering cities opportunities to reduce carbon emissions, traffic relief, and improve the quality of life for their residents (Chumin et al., 2021).

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 (Shu et al., 2013). 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 (Pal and Zhang, 2017; Raviv et al., 2013). The latter is often referred to as the Dynamic Bicycle Repositioning Problem (DBRP) (Kloimüllner et al., 2014; Mellou and Jaillet, 2019). 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 (Brinkmann et al., 2020; Legros, 2019; Li et al., 2021, 2018; Seo, 2020), the majority of the literature applies Mixed-integer Programming (MIP) models (Ghosh et al., 2019; Mellou and Jaillet, 2019; Zhang et al.,

* Corresponding author at: School of Management, Université du Québec à Montréal, 315 Rue Sainte-Catherine Est, Montréal, H2X 3X2, Canada.
E-mail addresses: jiaqi.liang@polymtl.ca (J. Liang), maria-clara.martins-silva@polymtl.ca (M.C.M. Silva), daniel.aloise@polymtl.ca (D. Aloise), jena.sanjay-dominik@uqam.ca (S.D. Jena).

https://doi.org/10.1016/j.ejtl.2024.100147
Received 23 February 2024; Received in revised form 3 October 2024; Accepted 4 November 2024
Available online 22 November 2024

2017; Zheng et al., 2021), 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 (Ghosh et al., 2019; Lowalekar et al., 2017; Shui and Szeto, 2018; You, 2019; Zhang et al., 2017, 2021), with both rental and return demands estimated either by naive predictions (e.g., the historical mean) (Ghosh et al., 2015, 2017; Lowalekar et al., 2017; Zhang et al., 2017) or more sophisticated Machine Learning (ML) techniques (Zamir, 2020; Zhang et al., 2021). However, minimizing lost demand may result in sub-optimal rebalancing solutions if the predicted rental and return are not accurate enough. Aiming at an easily understandable decision-making process that provides robustness to demand fluctuations, BSS operators, like BIXI Montreal, often use *inventory intervals* and *target inventories* within the rebalancing process. The inventory interval of a station refers to the range of its inventory at which it is considered balanced, ensuring that the station maintains a sufficient level of available bikes and free docks. The target inventory of a station represents the ideal number of bikes that would ensure optimal service. At BIXI Montreal, stations that find their inventories out of the defined interval raise an alert, based on which the operator may decide to rebalance that station.

While many methods have been proposed to compute inventory intervals and target inventories (see e.g., Huang et al., 2020; Hulot et al., 2018; Liu et al., 2016; Raviv and Kolka, 2013), only a few have incorporated them into optimization models. Notably, most relevant studies (Gleditsch et al., 2022; Kloimüllner et al., 2014; Schuijbroek et al., 2017) either focus on single-period models or minimize the deviation of target inventory at the end of the planning process only. Although Vogel (2016) and Vogel et al. (2014) 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., Ghosh et al., 2019; Gleditsch et al., 2022; Mellou and Jaillet, 2019; Zamir, 2020; Shui and Szeto, 2018), 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 a variety of these gaps, specifically: What are the benefits of using inventory intervals and targets, as opposed to minimizing lost demand? How does the accuracy of the demand prediction model impact the quality of the induced rebalancing decisions? And, what are the benefits of reoptimizing throughout the planning horizon?

Contributions. Answering the questions above requires the evaluation of the complete pipeline of an automatized and data-driven rebalancing process in BSSs. As such, 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. Rebalancing models with inventory intervals and targets

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 Liang et al., 2024): distance-based metrics, loading-based metrics, and demand-based metrics. Distance-based metrics are associated with the traveling distance of vehicles, mainly including traveling costs, traveling time, and fuel consumption (see e.g., Akova et al., 2022; Ghosh et al., 2017; Rainer-Harbach et al., 2015; Zheng et al., 2021). Loading-based metrics are associated with the number of handling (i.e., loading and unloading) operations (see e.g., Hu et al., 2021; Tang et al., 2020). 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., Ghosh et al., 2015; Hu et al., 2021; Kloimüllner et al., 2014; Mellou and Jaillet, 2019; Zhang et al., 2021).

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., Contardo et al., 2012; Ghosh et al., 2017, 2019; Hu et al., 2021; Lowalekar et al., 2017; Shui and Szeto, 2018; You, 2019; Zhang et al., 2017, 2021; Zheng et al., 2021). Concurrently, there have been a few attempts to minimize the deviation between the inventories of the stations and their target inventories (Gleditsch et al., 2022; Kloimüllner et al., 2014) or inventory intervals (Vogel et al., 2014), 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.

Table 1
Characterization of rebalancing optimization literature incorporating inventory target values or intervals.

Characteristic		Paper						This work
		Chemla et al. (2013)	Chen et al. (2023)	Gleditsch et al. (2022)	Kloimüllner et al. (2014)	Schuijbroek et al. (2017)	Vogel (2016)	
Planning horizon: multi-period			✓		✓		✓	✓
Solution approach: optimization model		✓	✓	✓		✓	✓	✓
Target type:	Target value	✓	✓	✓	✓			✓
	Inventory interval					✓	✓	✓
Target implementation: in OF				✓	✓		✓	✓
Target frequency: at each time-period							✓	✓
Performance evaluation via experiments			✓	✓				✓

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 inventory values represent specific inventory levels that operators aim to uphold at each station. Both are typically designed to ensure a high rate of demand satisfaction. We first review the literature that aims at incorporating such targets into planning problems. Then, approaches to compute either target values or inventory intervals are reviewed.

Inventory intervals and targets in optimization models. Even though inventory intervals and target inventories are often used concepts in the planning processes of BSS operators, only a small body of literature has incorporated them into optimization models. The problem variants considered in such works, as well as the way how target values and intervals are considered, may differ substantially.

Table 1 aims at visualizing the characteristics of the six papers identified to incorporate either inventory target values or inventory intervals within the context of BSS rebalancing planning. Specifically, half of the papers explicitly consider a multi-period planning horizon, which is desirable in a planning problem with such dynamic demand synergies. The other half focuses on single-period problems (which may also refer to overnight static rebalancing). Most papers present a mathematical optimization model, which potentially enables decision-makers to identify optimal planning solutions, whereas one paper only proposes a heuristic. Four papers focus on reaching a specific target value at each station inventory, while two consider target intervals. Target deviation is minimized within the objective function in half of the papers, while the other half aims at enforcing targets within constraints. Such latter option, however, may be too restrictive, or even render the model infeasible if the target values or intervals are ill defined. We therefore here consider that minimizing deviation from target values or intervals is a more appropriate choice. Next, only one work aims at reaching the proposed targets at each of the time-periods within a multi-period setting. The remainder either consider single-period models, or consider the target only for the last time-period of the planning horizon. BSS operators use inventory targets as a mean of preparing station inventories for the upcoming rental and return demand. As such, considering them at each of the time-periods is necessary. Finally, only two of the papers have provided empirical evaluations of the proposed approaches, which makes conclusions on their practical suitability difficult.

The model of Schuijbroek et al. (2017) comes closest to the proposition of our paper. The authors focus on target intervals, but within their computational experiments, the model optimizes different criteria and ignores the target intervals. As such, the authors do not evaluate the benefits of using inventory intervals. In our work, we aim at providing a systematic comparison of such approach, additionally including models that minimize deviation from target values. The other two multi-period models (Chen et al., 2023; Kloimüllner et al., 2014) consider target values only at the end of the planning horizon, which makes the models unsuitable for our purposes. Further, one of the models does not consider target values within the objective function, while the other does not provide a corresponding empirical performance analysis.

We therefore conclude a significant lack in the literature that simultaneously aims at satisfying either target values or inventory intervals at each of the time-periods within a multi-period setting. Our paper aims at filling this gap, and at providing a systematic empirical comparison of such approaches within different reoptimization modes.

Computation of inventory intervals and targets. 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 (Gammelli et al., 2022; Gleditsch et al., 2022; Huang et al., 2020; Raviv and Kolka, 2013). Inventory intervals have been computed in a similar fashion (Schuijbroek et al., 2017; Hulot et al., 2018), selecting those that minimize the likelihood of a station becoming either empty or full. Finally, different from those approaches, Datner et al. (2019) and Héctor et al. (2021) 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., Eren and Uz, 2020; Gallop et al., 2011; Gebhart and Noland, 2014; Hulot et al., 2018; Kim, 2018; Liu et al., 2016; O'Brien et al., 2022). To this end, Hulot et al. (2018) extend the notion of service-levels proposed by Schuijbroek et al. (2017), 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.

2.2. 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 (Vogel et al., 2011). Literature on demand prediction in BSSs can be divided into approaches predicting at global demand level (see e.g., Gebhart and Noland, 2014; Sathishkumar et al., 2020; Yin et al., 2012), at the level of station-clusters (see e.g., Borgnat et al., 2011; Feng et al., 2018; Vogel et al., 2011), and at the level of individual stations (see e.g., Boufidis et al., 2020; El-Assi et al., 2017; Hulot et al., 2018; Pan et al., 2019; Zamir, 2020). 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

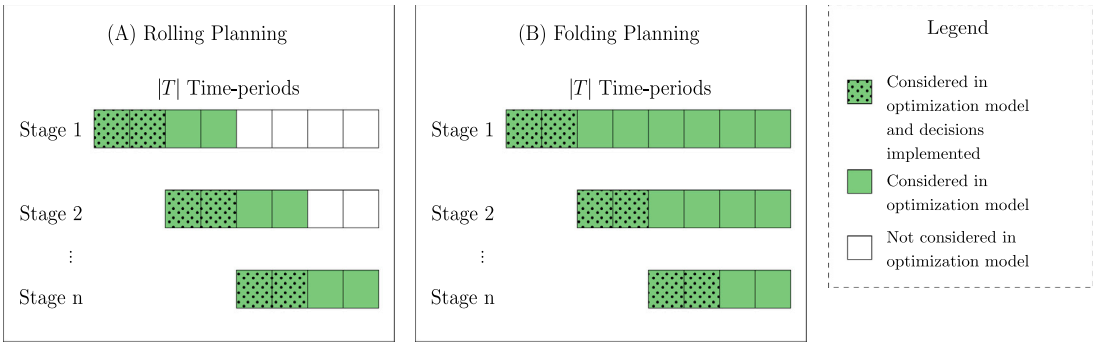

Fig. 1. The structure of rolling and folding planing.

to estimate future demand (Alvarez-Valdes et al., 2016; Datner et al., 2019; Ghosh et al., 2015; Gleditsch et al., 2022; Raviv and Kolka, 2013; Schuijbroek et al., 2017; Zhang et al., 2017), which can be seen as a naive predictor. Alternatively, rental and 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., Ghosh et al., 2017; Lowalekar et al., 2017; Zhang et al., 2017; Zheng et al., 2021). While such prediction methods centered on historical mean demand have been quite popular, several studies (see, e.g., Froehlich et al., 2009; Liu et al., 2016; Lozano et al., 2018) 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 (Boufidis et al., 2020; Hulot et al., 2018; Lozano et al., 2018; Wu et al., 2019; Yin et al., 2012). Typically, such models integrate weather and temporal features, highlighting their importance to accurately predict demand. Here, Hulot et al. (2018) 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.

2.3. 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., Ghosh et al., 2015; Kloimüllner et al., 2014; Lu, 2016; Sayarshad et al., 2012; Shu et al., 2013; Zhang et al., 2021) 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. Fig. 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

Table 2
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 $t \in T$.

implementing rolling planning for both multi-period (see, e.g., Ghosh et al., 2019; Lowalekar et al., 2017; Mellou and Jaillet, 2019; Zamir, 2020; Shui and Szeto, 2018; Zhang et al., 2017) and single-period models (see e.g., Ghosh et al., 2016; Gleditsch et al., 2022; Hu et al., 2021). Note that, in the case of rolling single-period models, the planning is essentially myopic. Rolling planning 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 Fig. 1(B). While this is generally a common approach in multi-period models (Lin et al., 2022), 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 a multi-period mixed-integer programming model provided by Liang et al. (2024), shown to provide consistently high performance among a large variety of different model variants.

The input parameters are listed in Table 2 and decision variables are shown in Table 3. 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

Table 3
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.

planning horizon with $|T|$ time-periods, where each time period $t \in T$ represents a duration of L_t minutes.

This formulation expresses the expected rental and return demand as input parameters $f_s^{+,t}$ and $f_s^{-,t}$, respectively. While the decision variables $x_s^{+,t}$ and $x_s^{-,t}$, separately capturing the rentals and returns, do not allow for keeping track of the origin–destination pair of trips, this modeling technique has been quite popular in the literature (see, e.g., Contardo et al., 2012; Kloimüllner et al., 2014; Lowalekar et al., 2017; You, 2019), given that it results in more compact models that are computationally more tractable. Indeed, Liang et al. (2024) conclude that such modeling technique results in planning solutions as good as models that explicitly consider the origin–destination pair in the trip demand and the corresponding decision variables.

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 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:

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

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

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

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

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

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

$$0 \le d_s^t \le C_s \qquad \forall\, s \in S \tag{7}$$

$$0 \le x_s^{+,t} \le f_s^{+,t}, 0 \le x_s^{-,t} \le f_s^{-,t} \qquad \forall\, s \in S,\ t \in T \tag{8}$$

$$0 \le r_{s,v}^{+,t}, r_{s,v}^{-,t} \le \hat{C}_v \qquad \forall\, s \in S,\ v \in V,\ t \in T \tag{9}$$

$$z_{s,v}^t \in \{0,1\} \qquad \forall\, s \in S,\ v \in V,\ t \in T. \tag{10}$$

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.

The above model, denoted as 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 **BSS** based on **T**arget Inventories (DROB-T) and DROB-I. The parameters and variables used in both models are depicted in Table 4.

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

$$\min \quad \sum_{s\in S}\sum_{t\in T} |\ell_s^t - d_s^t| \tag{11}$$

$$\text{s.t.} \quad (2)\text{–}(10).$$

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:

$$\min \quad \sum_{s\in S}\sum_{t\in T} e_s^{-,t} + e_s^{+,t} \tag{12}$$

$$\text{s.t.} \quad \underline{\ell}_s^t - e_s^{-,t} \le d_s^t \qquad \forall\, s \in S,\ t \in T \tag{13}$$

$$d_s^t \le \overline{\ell}_s^t + e_s^{+,t} \qquad \forall\, s \in S,\ t \in T \tag{14}$$

$$e_s^{-,t} \ge 0, e_s^{+,t} \ge 0 \qquad \forall\, s \in S,\ t \in T \tag{15}$$

$$(2)\text{–}(10).$$

Fig. 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, Fig. 2 also illustrates the deviation $|\ell_s^t - d_s^t|$ from the current target value. By

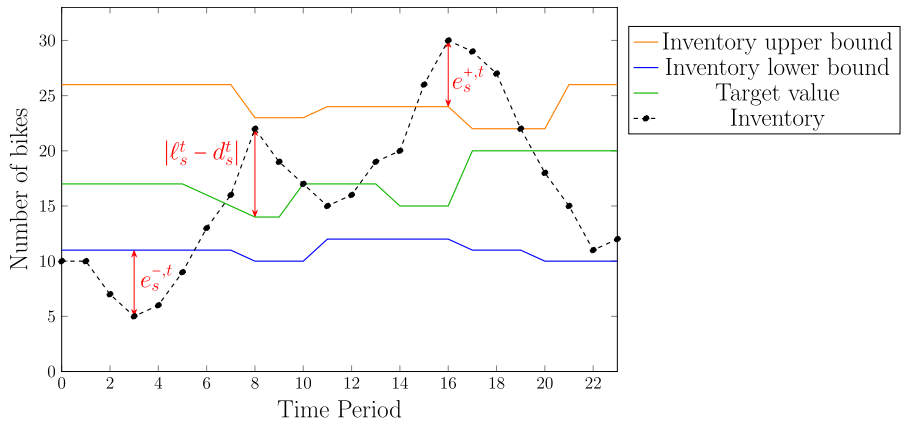

Fig. 2. Example of deviations from inventory intervals and target inventory.

Table 4
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 $t \in T$

Table 5
Comparative analysis of three objective functions for rebalancing operations.

Objective function	Example 1		Example 2	
	Dropped off bikes	Expected lost demand[a]	Dropped off bikes	Expected lost demand[a]
DROB-LD (1)	1	0.33	0	0.67
DROB-T (11)	8	0	5	0
DROB-I (12)	2	0	2	0

[a] 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.

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

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 5 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 h 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

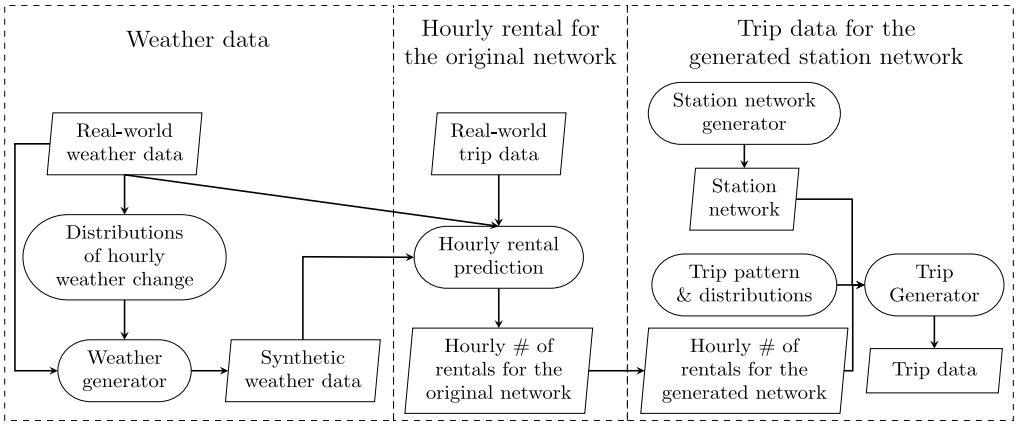

Fig. 3. Process to generate weather data, station network and trip data.

computational results, comparing the performance of the various planning models. Finally, Section 4.1.4 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.

The general data generation process of the here employed instance generator is depicted in Fig. 3. We extend the instance generator proposed by Liang et al. (2024) (corresponding to the third box in Fig. 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 Liang et al. (2024) 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 Fig. 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 Fig. 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., Eren and Uz, 2020; Gallop et al., 2011; Gebhart and Noland, 2014; Kim, 2018): 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 three ground truth problem instances, denoted GT1, GT2, and GT3, each of which contains a *station network*, as well as hourly weather data and detailed trip information for 500 days. In all 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

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

Table 6
Characteristics of the three considered ground truth instances.

Instance		GT1	GT2	GT3
Network (60 stations)	# of city centers	1	2	1
	# of stations per city center	9	6	9
	City center capacity	26%	35%	26%
Trip pattern	*OI*	32%	32%	20%
	OO	32%	32%	20%
	RD	23%	23%	40%
	RN	13%	13%	20%

non-work related trips occurring during the day (*RD* trips), and (iv) random non-work related trips occurring during the night (*RN* trips).

The characteristics of instance ground truths are described in Table 6. 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. In contrast, GT3 contains more irregular, random trips to emulate contexts where demand is less predictable.

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 Liang et al., 2024). 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 each ground truth 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 Hulot et al. (2018), 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 Liang et al., 2024).

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 Fig. 1(A). The *folding planning* uses all the remaining time-periods at each stage, as depicted in Fig. 1(B). The fine-grained discrete-event simulator from Liang et al. (2024) 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-min time-slots, which results in a particularly detailed and realistic simulation.

Computational Results. 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.

allows for the following observations:

1. **Comparison of proposed models.** From , models DROB-I and DROB-T generally outperform DROB-LD for all ground truths. 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 26.01% and 34.51%, respectively.)
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 Fig. 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 reduces the lost demand from 7.43% to 5.66% and 5.74%, respectively, over static planning. Note that updating the weather forecast may have a significant impact on the rolling planning, which will be discussed in Section 4.1.4. In terms of computing times, even though most models have been solved within 1-2 min, the folding planning requires much longer computing times for GT2.
3. **Comparison among ground truths.** While the general conclusions and tendencies are the same for all ground truths, 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. GT3 has more random trips, making demand prediction

Table 7
Results of dynamic rebalancing models for GT1, GT2, and GT3 under regular prediction.

Instance	Model	Reopt. mode	Opt. time (min)	Lost demand (%)					
				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
GT3	DROB-LD	*S*	0.52	18.99		6.34		13.35	
		R	0.03	18.08		6.27		12.78	
		F	0.53	17.98		6.14		12.66	
	DROB-I	*S*	0.01	18.11	▼ −4.63	5.79	▼ −8.68	12.58	▼ −5.77
		R	0.19	16.03	▼ −11.34	4.97	▼ −20.73	10.99	▼ −14.01
		F	0.08	16.03	▼ −10.85	4.87	▼ −20.68	10.94	▼ −13.59
	DROB-T	*S*	0.03	17.87	▼ −5.90	5.33	▼ −15.93	12.23	▼ −8.39
		R	13.10	15.39	▼ −14.88	4.77	▼ −23.92	10.53	▼ −17.61
		F	25.08	15.33	▼ −14.74	4.71	▼ −23.29	10.47	▼ −17.30

more difficult and resulting in higher lost demand. However, DROB-I and DROB-T still outperform DROB-LD, demonstrating the robustness of our two models even when trip patterns are less regular and predictable.

It can be observed that for DROB-I under GT2, and for DROB-T under GT3, the average computing times are rather high, exceeding 10-20 min on average. An analysis has shown that for those optimization runs, the MIP solver tends to find high-quality solutions quickly, but takes a long time to prove optimality. We therefore also report results with a 5-min time limit for these two specific cases (see Appendix C.1). The results indicate that the solution quality remains stable even under drastically reduced computing time resources, suggesting that our models remain effective and practically useful for time-sensitive planning contexts. Finally, we also report results from a study comparing the performance under 30 and 60-min time-period discretization in Appendix C.2. Such results confirm that, generally, more rebalancing operations translate into an improved lost demand, while DROB-I and DROB-T tend to outperform DROB-LD.

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 within the simulator. These experiments establish an empirical performance bound and provide insights into the efficiency of the optimization models and reoptimization modes. Fig. 4 presents the total lost demand obtained using the predicted trip demand (i.e., the same as used for) 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 prediction may obviously lead to reduced unmet demand, we will next investigate how those models perform when predictions are less accurate.

4.1.3. Analysis of planning solutions characteristics

In this section, we provide an analysis of several key performance indicators of the planning solutions suggested by the various models. Table 8 summarizes such results for GT1. Specifically, we report the average number of rebalanced bikes ("# of rebal. bikes") conducted per day and the average total daily travel distance (in km) covered by all 4 vehicles ("Travel distance"). Additionally, we present three metrics related to the travel patterns measured in percentage values: the proportion of time-periods at which a vehicle remains at the same station from one time-period to the next one ("Consecutive station stays"), which is indicative of the general relocation frequency of vehicles; the proportion of time-periods a vehicle visits stations located in a city-center during consecutive time-periods ("consecutive city center stays"), which tends to be close to each other, potentially indicating smaller travel distances; and the proportion of transitions between two stations where the travel time is estimated to be less than 20 min ("consecutive short visits"), generally indicating short travel distances.

Table 8 demonstrates that both DROB-I and DROB-T perform more rebalancing operations than DROB-LD, which is in line with our illustrative Example 1 in Section 3.2, allowing to more effectively reduce lost demand (see). Surprisingly, the increased number of rebalanced bikes does not incur larger traveling distances. This can be attributed

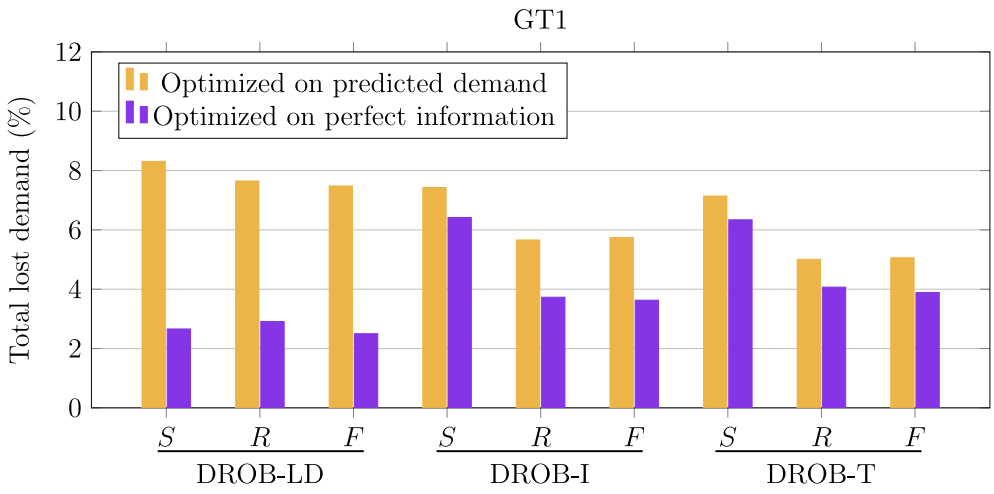

Fig. 4. Results of total lost demand for GT1 using regular prediction and ground truth.

Table 8
Rebalancing operation and travel distance statics for GT1 under regular prediction.

Model	Reopt. mode	# of rebal. bikes	Travel distance	Consecutive station stays	Consecutive city center stays	Consecutive short visits
DROB-LD	*S*	99.03	160.92	0.97%	15.47%	43.60%
	R	110.41	162.68	1.47%	14.13%	45.93%
	F	107.18	164.88	1.27%	14.00%	45.80%
DROB-I	*S*	215.42	139.76	8.03%	10.47%	54.23%
	R	279.71	154.76	1.67%	12.20%	46.10%
	F	284.19	153.24	1.70%	11.10%	46.17%
DROB-T	*S*	143.00	123.04	3.17%	11.37%	65.23%
	R	254.79	151.92	0.03%	15.33%	46.73%
	F	253.41	150.27	0.10%	14.03%	48.33%

to several factors, as indicated by the three travel pattern percentage metrics. DROB-I has the highest percentage of consecutive station stays, which suggests that vehicles remain at the same station more frequently, leading to less travel distance. DROB-T shows the highest percentage of short visits and a relatively high percentage of city center stays, indicating more frequent vehicle movements to nearby stations, thereby reducing overall travel distance. One can conclude that DROB-LD tends to relocate more often, but relocates fewer bikes. In contrast, DROB-I and DROB-D relocate less often, but relocate more bikes at each station. Even though none of the models explicitly considers travel distances in their objective functions, the results above are not surprising when considering how rebalancing operations are triggered in each model. DROB-LD is more sensitive to demand prediction, as it explicitly considers lost demand in its objective function. Anticipated lost demand tends to be quite variable among the stations, giving incentives to the model to relocate to a station where lost demand can be reduced slightly more than at the current station. In contrast, DROB-I and DROB-T are less sensitive to demand prediction, where stations with similar demand levels may share similar inventory intervals/targets. In other words, the loss-function considered in such models may indicate a similar urgency for many stations, reducing the need to relocate to a different station.

We may conclude that the two new models, DROB-I and DROB-T, tend to provide a more sober assessment of relocation urgency among the different stations, implicitly acknowledging the uncertainty of demand fluctuations, and naturally providing less costly rebalancing routes. This being said, we reiterate that none of the models explicitly considers the relocation distances in their objective function, as the focus of our study lies in minimizing lost demand under vehicle resource constraints. If economic efficiency is desired, such an indicator should be explicitly modeled, either as a penalty within the objective function or by constraining relocations to sufficiency small distances.

4.1.4. 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 h 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.

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 Fig. 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 the static planning, as in the previous experiments, optimizes only once and that for the entire planning horizon. For all three reoptimization

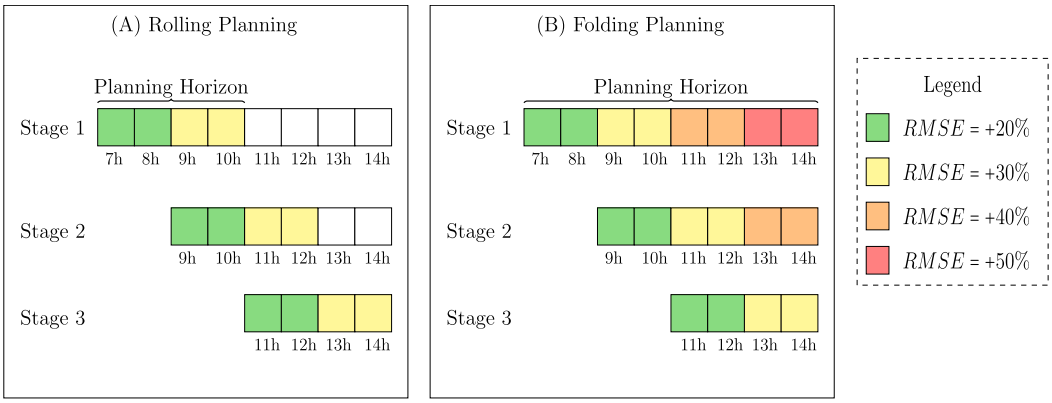

Fig. 5. Noise added to the demand prediction over the planning horizon of the rolling and folding planning.

Table 9
Results of dynamic rebalancing models for GT1 and GT2 under underestimating predictions.

Instance	Model	Reopt. mode	Opt. time (min)	Lost demand (%)					
				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	8.60	▼ −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

modes, demand predictions for later time-periods deteriorate, i.e., have a higher level of noise. In other words, the first 2 time-periods have a noise corresponding to an RMSE increase of 20%, the next time periods an RMSE increase of 40%, and so on. The visual illustration of the static planning would therefore correspond to the planning horizon shown for stage 1 in Fig. 5(B), not for stages 2 or 3.

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

Based on , 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 consistently 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). 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 Fig. 6. Indeed, overestimating predictions results in more rebalancing operations to meet the high demand. The respective statistics for DROB-I and DROB-T can be found in Appendix C.3. 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.

Remarks. Overall, the results indicate that DROB-LD is more sensitive to demand prediction accuracy than DROB-I and DROB-T, as DROB-LD explicitly considers predicted demand 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 Fig. 7, and between static and folding planning in Fig. 8 for both GT1 and GT2. It is

Table 10
Results of dynamic rebalancing models for GT1 and GT2 under overestimating predictions.

Instance	Model	Reopt. mode	Opt. time (min)	Lost demand (%)					
				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

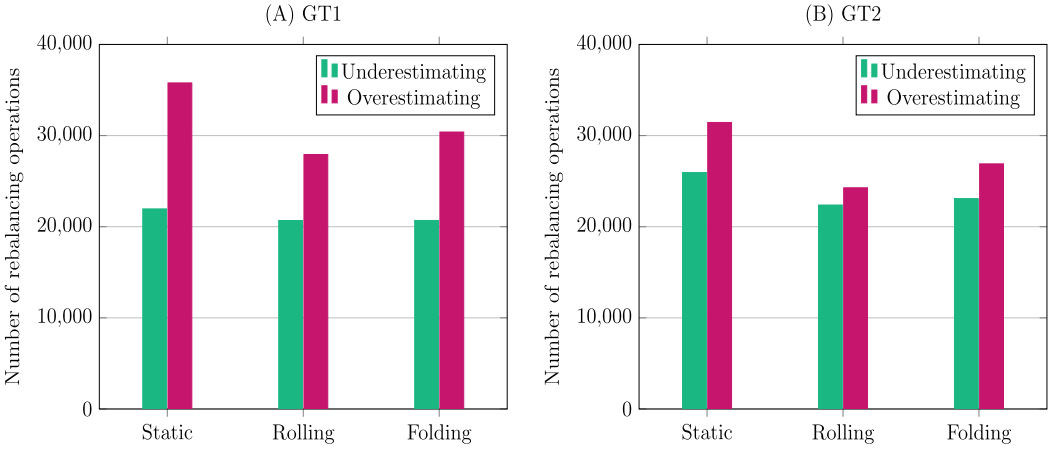

Fig. 6. Number of rebalancing operations in GT1 and GT2 for the DROB-LD.

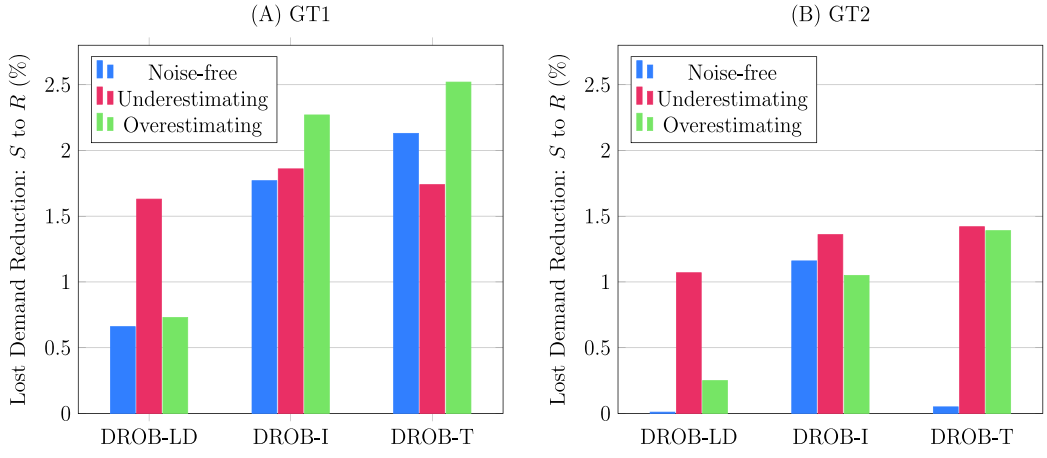

Fig. 7. Improvement of the total lost demand from static to rolling planning.

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

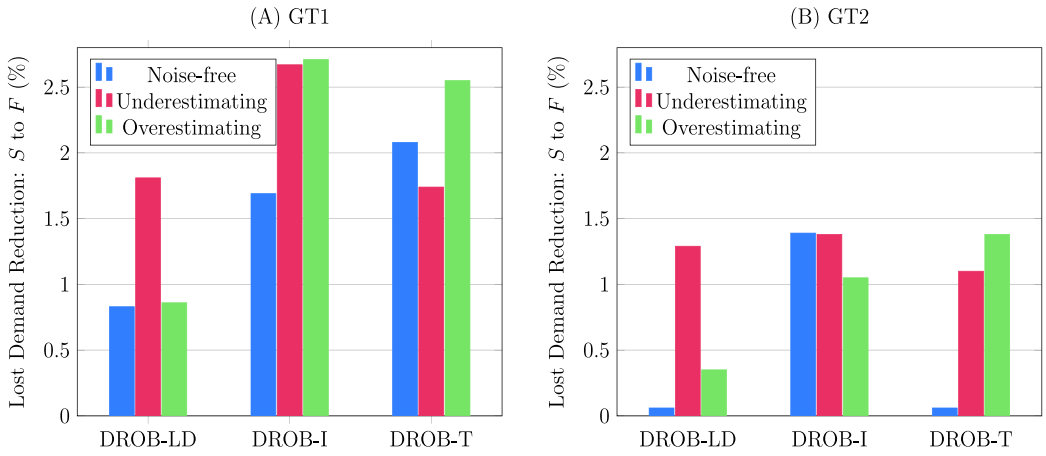

Fig. 8. Improvement of the total lost demand from static to folding planning.

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 Fig. 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 (0 h–23 h), 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.

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. Rebalancing is then performed within a specific cluster or between different clusters (see, e.g., Calafiore et al., 2019; Forma et al., 2015; Ghosh et al., 2015; Huang et al., 2022; Jin et al., 2022; Liu et al., 2016).

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

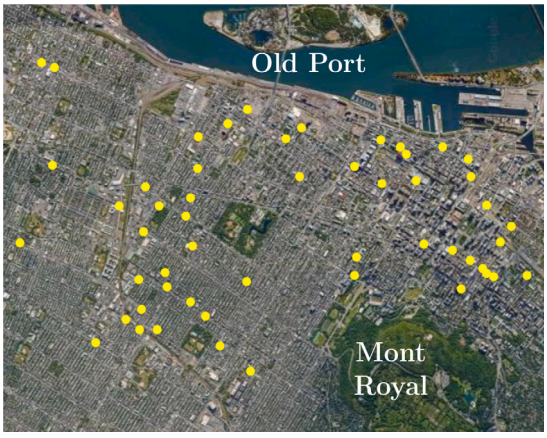

Fig. 9. Cluster of BIXI stations located in Montreal (Canada).

We follow the approach of Liang et al. (2024) to cluster the stations according to their trip behavior 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 Fig. 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.

The results are visualized in Fig. 10. Detailed results for each day can be found in Appendix D. 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 successful 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

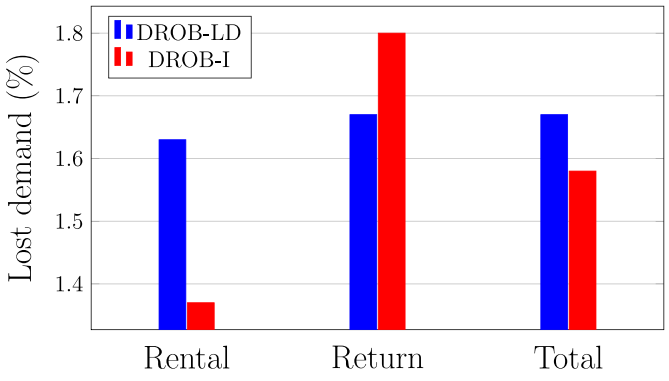

Fig. 10. Lost demand of DROB-I and DROB-LD on a cluster from BIXI.

the station inventories and reducing the total lost demand. In terms of computing time, both models can be solved to optimality within 1 min.

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 explores 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 34.51%, while DROB-I decreases lost demand by up to 28.74%. 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 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 in 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 may be a promising research direction, which is likely to be highly useful for both academia and practitioners to approach rebalancing in large-scale station networks. Mathematical decomposition methods are likely to be particularly promising avenues. For instance, Branch-and-cut-and-price algorithms for Pick-up-and-delivery problems (see, e.g., Rostami et al., 2021) may be extended to the here considered dynamic rebalancing problem, and Benders decomposition methods for static rebalancing (see, e.g., Dell'Amico et al., 2018) may be extended to the dynamic case.

CRediT authorship contribution statement

Jiaqi Liang: Data curation, Formal analysis, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Maria Clara Martins Silva:** Conceptualization, Data curation, Formal analysis, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Daniel Aloise:** Conceptualization, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Supervision, Validation, Writing – original draft, Writing – review & editing. **Sanjay Dominik Jena:** Conceptualization, Formal analysis, Funding acquisition, Investigation, Methodology, Software, Supervision, Validation, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors are thankful to BIXI Montreal, in particular, Nicolas Blain, 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 authors 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).

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 Hulot et al. (2018). 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

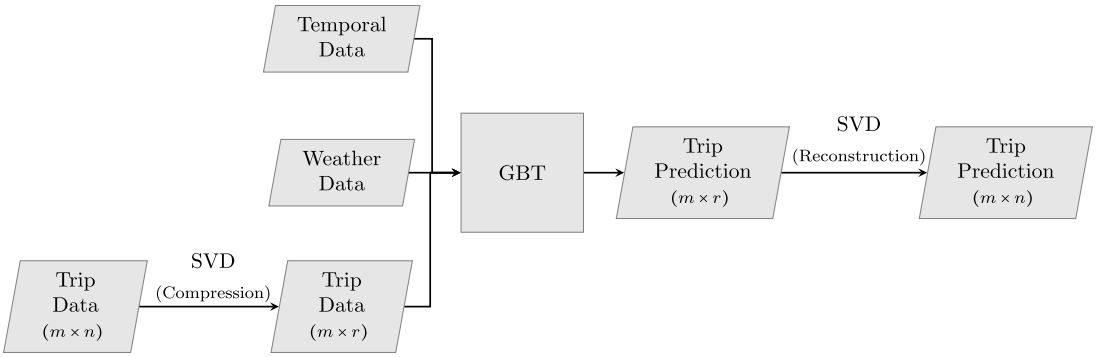

Fig. A.1. Pipeline of the predictive model.

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. Fig. A.1 illustrates the model's pipeline.

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

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

where $\mathcal{L} = \min\{i \in \{0,\dots,C_s\} | SL_s^t(i) \ge \Omega_s^t\}$, and $\mathcal{U} = \max\{i \in \{0,\dots,C_s\} | SL_s^t(i) \ge \Omega_s^t\}$. 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 Fig. 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 Fig. B.1. The overlaid red curves illustrate the normal distributions in which the parameters are computed using the Maximum Likelihood Estimator.

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

Appendix C. Supplementary experimental results on synthetic data

C.1. Results with shorter computing time limit

In this section, we present results for configurations of ground truths and model variants, where the original average computing times (as reported in) exceed 10 min. The results are summarized in , indicating that despite the reduced computing time resources, the solution quality does not significantly deteriorate. These findings reinforce the efficiency of our models, demonstrating their ability to provide near-optimal solutions within a short time frame, making them suitable for practical, time-sensitive applications.

C.2. Time discretization

In this section, we report experimental results with 60 and 30-min time-periods in . Note that a shorter planning horizon of 4 h is used, rather than the original 8 h, due to the computational resources required, as the models take too much time to solve to optimality and may risk running out of memory as the branching tree becomes too extensive within the computing time limit of 24 h.

It can be observed that using shorter time-periods with more frequent rebalancing operations can effectively reduce lost demand. For example, the lost demand for DROB-I under the rolling planning decreases from 5.03% to 3.57%. Additionally, DROB-I and DROB-T continue to outperform DROB-LD, further reinforcing the effectiveness of our approach.

C.3. Rebalancing operations with noisy prediction

We further report the total number of rebalancing operations obtained from DROB-I and DROB-T in Figs. C.2 and C.3, respectively. When comparing with Fig. 6, one can observe that DROB-I and DROB-T

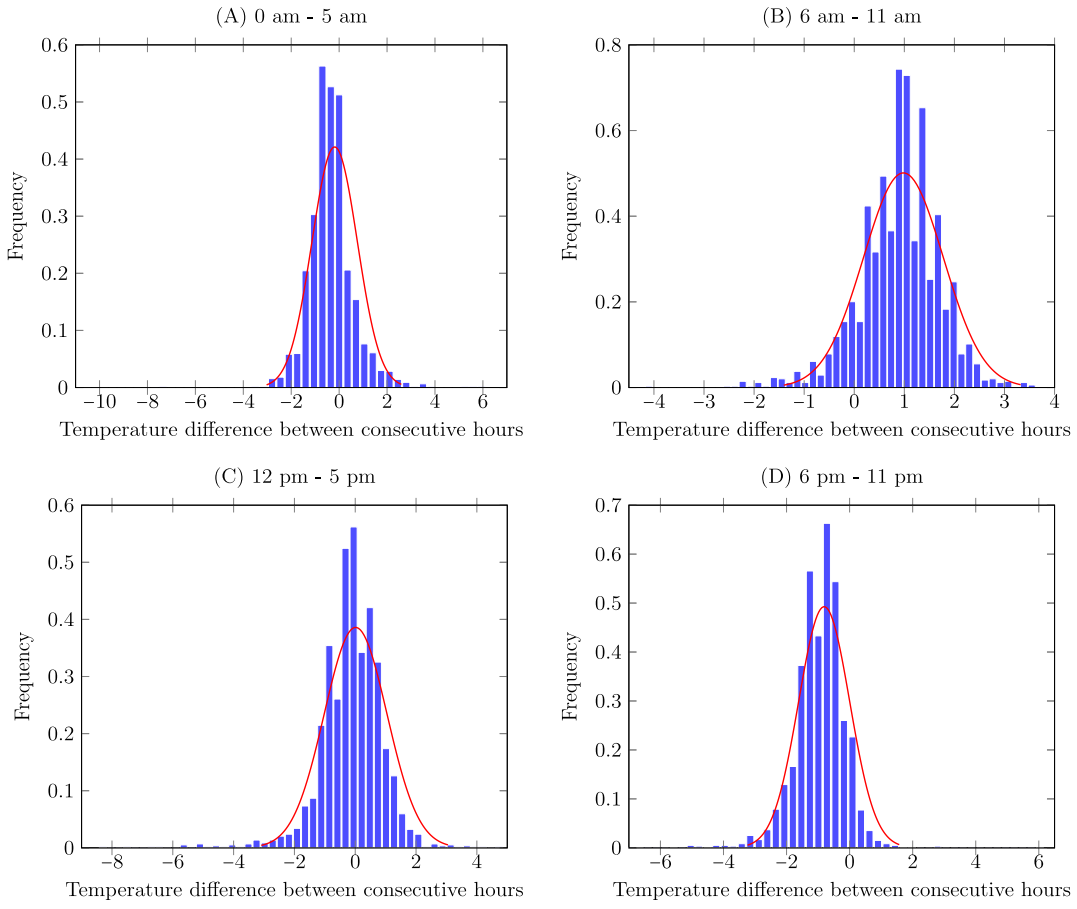

Fig. B.1. Distributions of temperature changes between consecutive hours for four time-periods.

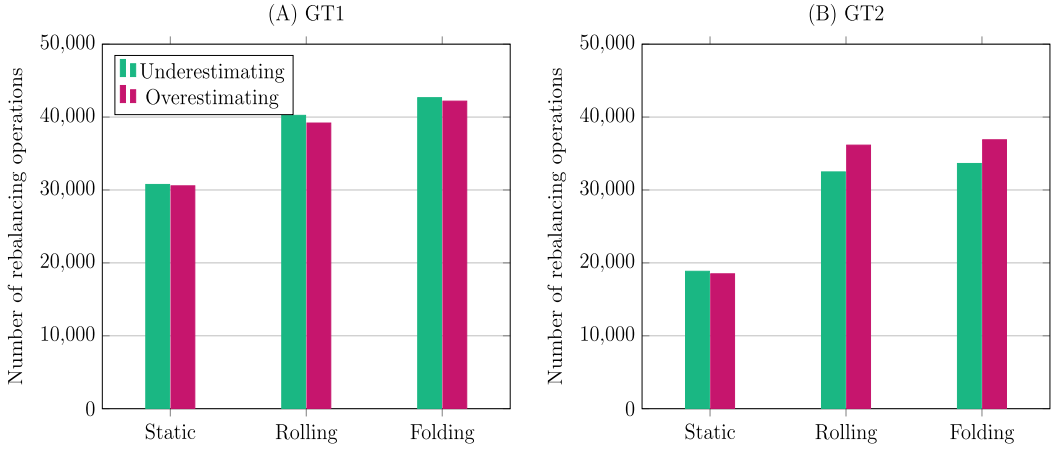

Fig. C.2. Number of rebalancing operations in GT1 and GT2 for DROB-I.

Table C.1
Results with a time limit of 5 min.

Instance	Model	Reopt. mode	Opt. time (min)	Lost demand (%)					
				Rental	Δ (%)	Return	Δ (%)	Total	Δ (%)
GT2	DROB-I	*S*	0.04	7.76	▼ −6.17	2.51	▲ 49.40	5.24	▲ 2.14
		R	3.04	6.31	▼ −22.67	1.35	▼ −19.16	3.92	▼ −22.53
		F	4.12	6.27	▼ −22.69	1.30	▼ −19.75	3.87	▼ −22.75
GT3	DROB-T	*S*	0.03	17.87	▼ −5.90	5.33	▼ −15.93	12.23	▼ −8.39
		R	2.44	15.39	▼ −14.88	4.77	▼ −23.92	10.53	▼ −17.61
		F	2.69	15.33	▼ −14.74	4.70	▼ −23.45	10.47	▼ −17.30

Table C.2
Results of dynamic rebalancing models for GT1 under different time period.

Length of time-period	Model	Reopt. mode	Opt. time (min)	Lost demand (%)					
				Rental	Δ (%)	Return	Δ (%)	Total	Δ (%)
60 min	DROB-LD	*S*	0.01	10.62		1.43		6.28	
		R	0.01	8.18		1.68		5.07	
		F	0.02	7.35		1.67		4.62	
	DROB-I	*S*	0.01	7.91	▼ −25.52	1.90	▲ 32.87	5.03	▼ −19.90
		R	0.01	6.95	▼ −15.04	1.80	▲ 7.14	4.48	▼ −11.64
		F	0.02	6.15	▼ −16.33	1.85	▲ 10.78	4.07	▼ −11.90
	DROB-T	*S*	0.01	7.17	▼ −32.49	1.91	▲ 33.57	4.64	▼ −26.11
		R	0.01	7.10	▼ −13.20	1.11	▼ −33.93	4.22	▼ −16.77
		F	0.02	6.87	▼ −6.53	1.21	▼ −27.54	4.15	▼ −10.17
30 min	DROB-LD	*S*	0.09	6.41		1.32		3.95	
		R	0.02	4.78		1.34		3.11	
		F	0.11	4.44		1.28		2.90	
	DROB-I	*S*	0.02	5.49	▼ −14.35	1.54	▲ 16.67	3.57	▼ −9.62
		R	2.98	4.44	▼ −7.11	1.26	▼ −5.97	2.89	▼ −7.07
		F	1.05	4.36	▼ −1.80	1.32	▲ 3.13	2.88	▼ −0.69
	DROB-T	*S*	0.05	5.42	▼ −15.44	1.41	▲ 6.82	3.48	▼ −11.90
		R	0.01	2.04	▼ −57.32	3.01	▲ 124.63	2.53	▼ −18.65
		F	0.30	5.42	▲ 22.07	1.39	▲ 8.59	3.47	▲ 19.66

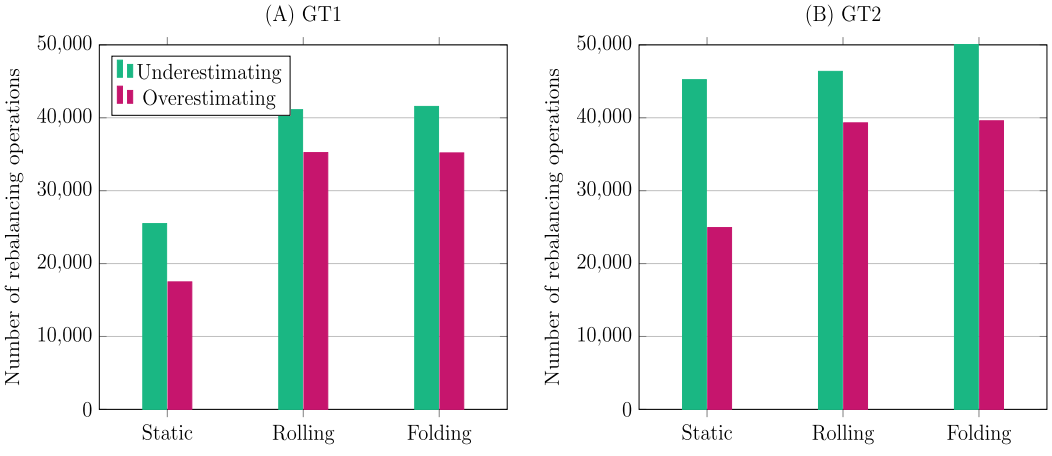

Fig. C.3. Number of rebalancing operations in GT1 and GT2 for DROB-T.

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

Day	Model	Opt. time	MIP	Lost demand (%)		
		(min)	gap (%)	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

tend to carry out more rebalancing operations than DROB-LD, for both underestimating and overestimating predictions. Furthermore, in contrast to the case of DROB-LD in Fig. 6, overestimating predictions do not necessarily lead to more rebalancing operations than underestimating predictions, indicating that DROB-I and DROB-T are less sensitive to prediction noise.

Appendix D. 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 D.1.

References

Akova, H., Hulagu, S., Celikoglu, H.B., 2022. Static bike repositioning problem with heterogeneous distribution characteristics in bike sharing systems. Transp. Res. Procedia 62, 205–212.

Alvarez-Valdes, R., Belenguer, J.M., Benavent, E., Bermudez, J.D., Muñoz, F., Vercher, E., Verdejo, F., 2016. Optimizing the level of service quality of a bike-sharing system. Omega 62, 163–175.

Borgnat, P., Abry, P., Flandrin, P., Robardet, C., Rouquier, J.-B., Fleury, E., 2011. Shared bicycles in a city: A signal processing and data analysis perspective. Adv. Complex Syst. 14 (03), 415–438.

Boufidis, N., Nikiforiadis, A., Chrysostomou, K., Aifadopoulou, G., 2020. Development of a station-level demand prediction and visualization tool to support bike-sharing systems' operators. Transp. Res. Procedia 47, 51–58.

Brinkmann, J., Ulmer, M.W., Mattfeld, D.C., 2020. The multi-vehicle stochastic-dynamic inventory routing problem for bike sharing systems. Bus. Res. 13 (1), 69–92.

Calafiore, G.C., Bongiorno, C., Rizzo, A., 2019. A robust MPC approach for the rebalancing of mobility on demand systems. Control Eng. Pract. 90, 169–181.

Chemla, D., Meunier, F., Calvo, R.W., 2013. Bike sharing systems: Solving the static rebalancing problem. Discrete Optim. 10 (2), 120–146.

Chen, Q., Fu, C., Zhu, N., Ma, S., He, Q.-C., 2023. A target-based optimization model for bike-sharing systems: From the perspective of service efficiency and equity. Transp. Res. B 167, 235–260.

Chumin, Y., O'Brien, O., DeMaio, P., Rabello, R., Chou, S., Benicchio, T., 2021. The Meddin bike-sharing world map report. https://bikesharingworldmap.com/reports/bswm_mid2021report.pdf. (Accessed 04 August 2023).

Contardo, C., Morency, C., Rousseau, L.-M., 2012. Balancing a Dynamic Public Bike-Sharing System, vol. 4, Cirrelt Montreal, Canada.

Datner, S., Raviv, T., Tzur, M., Chemla, D., 2019. Setting inventory levels in a bike sharing network. Transp. Sci. 53 (1), 62–76.

Dell'Amico, M., Iori, M., Novellani, S., Subramanian, A., 2018. The bike sharing rebalancing problem with stochastic demands. Transp. Res. B 118, 362–380.

El-Assi, W., Salah Mahmoud, M., Nurul Habib, K., 2017. Effects of built environment and weather on bike sharing demand: a station level analysis of commercial bike sharing in toronto. Transportation 44 (3), 589–613.

Eren, E., Uz, V.E., 2020. A review on bike-sharing: The factors affecting bike-sharing demand. Sustain. Cities Soc. 54, 101882.

Feng, S., Chen, H., Du, C., Li, J., Jing, N., 2018. 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, pp. 829–836.

Forma, I.A., Raviv, T., Tzur, M., 2015. A 3-step math heuristic for the static repositioning problem in bike-sharing systems. Transp. Res. B 71, 230–247.

Froehlich, J.E., Neumann, J., Oliver, N., 2009. Sensing and predicting the pulse of the city through shared bicycling. In: Twenty-First International Joint Conference on Artificial Intelligence. pp. 1420–1426.

Gallop, C., Tse, C., Zhao, J., 2011. A seasonal autoregressive model of vancouver bicycle traffic using weather variables. i-Manager's J. Civ. Eng. 1 (4), 9.

Gammelli, D., Wang, Y., Prak, D., Rodrigues, F., Minner, S., Pereira, F.C., 2022. Predictive and prescriptive performance of bike-sharing demand forecasts for inventory management. Transp. Res. C 138, 103571.

Gebhart, K., Noland, R.B., 2014. The impact of weather conditions on bikeshare trips in Washington, DC. Transportation 41 (6), 1205–1225.

Ghosh, S., Koh, J.Y., Jaillet, P., 2019. Improving customer satisfaction in bike sharing systems through dynamic repositioning. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19. pp. 5864–5870.

Ghosh, S., Trick, M., Varakantham, P., 2016. 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, pp. 3096–3102.

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

Ghosh, S., Varakantham, P., Adulyasak, Y., Jaillet, P., 2017. Dynamic repositioning to reduce lost demand in bike sharing systems. J. Artificial Intelligence Res. 58, 387–430.

Gleditsch, M.D., Hagen, K., Andersson, H., Bakker, S.J., Fagerholt, K., 2022. A column generation heuristic for the dynamic bicycle rebalancing problem. European J. Oper. Res. In press.

Héctor, G., Rafael, L., Ramirez-Nafarrate, A., 2021. A simulation-optimization study of the inventory of a bike-sharing system: The case of Mexico city ecobici's system. Case Stud. Transp. Policy 9 (3), 1059–1072.

Hu, R., Zhang, Z., Ma, X., Jin, Y., 2021. Dynamic rebalancing optimization for bike-sharing system using priority-based MOEA/D algorithm. IEEE Access 9, 27067–27084.

Huang, J., Sun, H., Li, H., Huang, L., Li, A., Wang, X., 2020. Central station-based demand prediction for determining target inventory in a bike-sharing system. Comput. J. 65 (3), 573–588.

Huang, J., Tan, Q., Li, H., Li, A., Huang, L., 2022. Monte carlo tree search for dynamic bike repositioning in bike-sharing systems. Appl. Intell. 1–16.

Hulot, P., Aloise, D., Jena, S.D., 2018. 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. pp. 378–386.

Jin, Y., Ruiz, C., Liao, H., 2022. A simulation framework for optimizing bike rebalancing and maintenance in large-scale bike-sharing systems. Simul. Model. Pract. Theory 115, 102422.

Kim, K., 2018. Investigation on the effects of weather and calendar events on bike-sharing according to the trip patterns of bike rentals of stations. J. Transp. Geogr. 66, 309–320.

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

Legros, B., 2019. Dynamic repositioning strategy in a bike-sharing system; how to prioritize and how to rebalance a bike station. European J. Oper. Res. 272 (2), 740–753.

Li, G., Cao, N., Zhu, P., Zhang, Y., Zhang, Y., Li, L., Li, Q., Zhang, Y., 2021. Towards smart transportation system: A case study on the rebalancing problem of bike sharing system based on reinforcement learning. J. Organ. End User Comput. (JOEUC) 33 (3), 35–49.

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

Liang, J., Jena, S.D., Lodi, A., 2024. Dynamic rebalancing optimization for bike-sharing systems: A modeling framework and empirical comparison. European J. Oper. Res. 317 (3), 875–889.

Lin, S., Chen, F.Y., Li, Y., Shen, Z.-J.M., 2022. 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).

Liu, J., Sun, L., Chen, W., Xiong, H., 2016. 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. pp. 1005–1014.

Lowalekar, M., Varakantham, P., Ghosh, S., Jena, S.D., Jaillet, P., 2017. Online repositioning in bike sharing systems. In: Proceedings of the International Conference on Automated Planning and Scheduling. Vol. 27, pp. 200–208.

Lozano, Á., De Paz, J.F., Villarrubia Gonzalez, G., Iglesia, D.H.D.L., Bajo, J., 2018. Multi-agent system for demand prediction and trip visualization in bike sharing systems. Appl. Sci. 8 (1), 67.

Lu, C.-C., 2016. Robust multi-period fleet allocation models for bike-sharing systems. Netw. Spat. Econ. 16 (1), 61–82.

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

O'Brien, O., DeMaio, P., Rabello, R., Chou, S., Benicchio, T., 2022. The meddin bike-sharing world map report. https://bikesharingworldmap.com/reports/bswm_mid2022report.pdf. (Accessed 04 August 2023).

Pal, A., Zhang, Y., 2017. Free-floating bike sharing: Solving real-life large-scale static rebalancing problems. Transp. Res. C 80, 92–116.

Pan, Y., Zheng, R.C., Zhang, J., Yao, X., 2019. Predicting bike sharing demand using recurrent neural networks. Proc. Comput. Sci. 147, 562–566.

Rainer-Harbach, M., Papazek, P., Raidl, G.R., Hu, B., Kloimüllner, C., 2015. PILOT, GRASP, and VNS approaches for the static balancing of bicycle sharing systems. J. Global Optim. 63 (3), 597–629.

Raviv, T., Kolka, O., 2013. Optimal inventory management of a bike-sharing station. IIE Trans. 45 (10), 1077–1093.

Raviv, T., Tzur, M., Forma, I.A., 2013. Static repositioning in a bike-sharing system: models and solution approaches. EURO J. Transp. Logist. 2 (3), 187–229.

Rostami, B., Desaulniers, G., Errico, F., Lodi, A., 2021. Branch-price-and-cut algorithms for the vehicle routing problem with stochastic and correlated travel times. Oper. Res. 69 (2), 436–455.

Sathishkumar, V., Jangwoo, P., Yongyun, C., 2020. Using data mining techniques for bike sharing demand prediction in metropolitan city. Comput. Commun. 153, 353–366.

Sayarshad, H., Tavassoli, S., Zhao, F., 2012. A multi-periodic optimization formulation for bike planning and bike utilization. Appl. Math. Model. 36 (10), 4944–4951.

Schuijbroek, J., Hampshire, R.C., Van Hoeve, W.-J., 2017. Inventory rebalancing and vehicle routing in bike sharing systems. European J. Oper. Res. 257 (3), 992–1004.

Seo, Y., 2020. A Dynamic Rebalancing Strategy in Public Bicycle Sharing Systems Based on Real-Time Dynamic Programming and Reinforcement Learning (Ph.D. thesis). Doctoral dissertation. Seoul National University, South Korea.

Shu, J., Chou, M.C., Liu, Q., Teo, C.-P., Wang, I.-L., 2013. Models for effective deployment and redistribution of bicycles within public bicycle-sharing systems. Oper. Res. 61 (6), 1346–1359.

Shui, C., Szeto, W., 2018. Dynamic green bike repositioning problem–a hybrid rolling horizon artificial bee colony algorithm approach. Transp. Res. D 60, 119–136.

Tang, Q., Fu, Z., Zhang, D., Guo, H., Li, M., 2020. Addressing the bike repositioning problem in bike sharing system: a two-stage stochastic programming model. Sci. Program. 2020, 1–12.

Vogel, P., 2016. Service network design of bike sharing systems. In: Service Network Design of Bike Sharing Systems: Analysis and Optimization. Springer International Publishing, Cham, pp. 113–135.

Vogel, P., Greiser, T., Mattfeld, D.C., 2011. Understanding bike-sharing systems using data mining: Exploring activity patterns. Proc.-Soc. Behav. Sci. 20, 514–523.

Vogel, P., Neumann Saavedra, B.A., Mattfeld, D.C., 2014. 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, pp. 16–29.

Wu, X., Lyu, C., Wang, Z., Liu, Z., 2019. Station-level hourly bike demand prediction for dynamic repositioning in bike sharing systems. In: Smart Transportation Systems 2019. Springer, pp. 19–27.

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

You, P.-S., 2019. A two-phase heuristic approach to the bike repositioning problem. Appl. Math. Model. 73, 651–667.

Zamir, K.R., 2020. Dynamic Repositioning for Bikesharing Systems (Ph.D. thesis). University of Maryland, College Park.

Zhang, J., Meng, M., Wong, Y.D., Ieromonachou, P., Wang, D.Z., 2021. A data-driven dynamic repositioning model in bicycle-sharing systems. Int. J. Prod. Econ. 231, 107909.

Zhang, D., Yu, C., Desai, J., Lau, H., Srivathsan, S., 2017. A time-space network flow approach to dynamic repositioning in bicycle sharing systems. Transp. Res. B 103, 188–207.

Zheng, X., Tang, M., Liu, Y., Xian, Z., Zhuo, H.H., 2021. Repositioning bikes with carrier vehicles and bike trailers in bike sharing systems. Appl. Sci. 11 (16), 7227.