CIRRELT

CIRRELT-2022-16

Data-driven Prioritization Strategies for Inventory Rebalancing in Bike-sharing systems

Maria Clara Martins Silva
Daniel Aloise
Sanjay Dominik Jena

June 2022

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

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

Data-driven Prioritization Strategies for Inventory Rebalancing in Bike-sharing systems

Maria Clara Martins Silva[1], Daniel Aloise[1,*], Sanjay Dominik Jena[2]

1. Computer Engineering and Software Engineering Department, Polytechnique Montreal
2. Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT) and Département d'analytique, opérations et technologies de l'information (AOTI), ESG, UQAM

Abstract. The popularity of bike-sharing systems has constantly increased throughout the last years, given their convenience for users, low usage costs, health benefits and at their contribution to environmental relief. However, satisfying all user demands remains a challenge, given that the inventory of bike-sharing stations tends to be unbalanced over time. Bike-sharing system operators therefore explicitly rebalance station inventories in order to provide both available bikes and empty docks to the commuters. In most systems, the operator manually selects the stations and amounts of bikes to be rebalanced among those that are considered unbalanced. In practice, such manual planning is likely to result in suboptimal system performance. In this paper, we propose three variants of a machine learning-based algorithm to select the stations that should be prioritized for rebalancing, using features such as the predicted trip demand, as well as the inventory levels at the stations themselves and their surrounding stations. We evaluate the performance of these prioritization strategies by simulating real-world trips using data from 2019 and 2020, each of which exhibits distinct travel patterns given the restrictive measures implemented in 2020 to prevent the spread of COVID-19. One of the strategies significantly improves the system's performance, reducing the lost demand up to 22% and the required rebalancing operations up to 12% when compared to the prioritization scheme currently used in practice. Finally, another strategy encourages the selection of stations that are geographically clustered, which may facilitate rebalancing operations afterwards.

Keywords: Bike-sharing, demand prediction, machine learning, inventory management.

Acknowledgements. The authors are grateful to BIXI-Montreal who provided resources throughout the project. The authors also thank the Natural Sciences and Engineering Research Council of Canada (NSERC) for its financial support.

---

* Corresponding author: daniel.aloise@polymtl.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2022

1. Introduction

Demand for bike-sharing systems (BSS) has constantly increased over the recent years, as they continue to provide various advantages: they are typically simple to use and do not require previous reservation; they have been shown to be an environmentally friendly transportation mode by reducing the ever-increasing amount of cars in circulation (Wang and Zhou, 2017); and, they contribute to a healthy lifestyle (Pucher et al., 2010) of the participants. Particularly throughout the ongoing pandemic, BSSs have been considered a transportation alternative with particularly low risk of user contamination (Pase et al., 2020).

In this paper, we focus on dock-based BSSs, in which stations are located in different parts of the city, and from which commuters may rent and return bikes. While dock-based systems have several advantages (e.g. users get used to the location of stations and bikes), a main issue is that the station inventories may quickly become unbalanced, i.e., either rental demand cannot be met, given that not a sufficient number of bikes is available, or return demand cannot be met, when the station has no empty docks. Such imbalances often occur during rush hours on weekdays, when commuters relocate from their residential areas to the areas they work in the morning, and do the return trip in the afternoon (Mellou and Jaillet, 2019). Unmet user demand likely causes user dissatisfaction, which the system operators seeks to avoid as best as possible, given that it ultimately reduces the user base as the system's reputation is damaged.

An effective way to combat station inventory imbalances is to redistribute bikes among stations, a process known as *rebalancing*. The literature distinguishes two main types of rebalancing. User-based rebalancing consists of incentives given to the users in order to return bikes at stations before they become empty (Vallez et al., 2021). Operator-based rebalancing is carried out by the BSS operators themselves, typically by dispatching trucks that relocate bikes between the stations. In this work, we focus on operator-based rebalancing, which has shown to be effective to increase demand satisfaction (see, e.g. DeMaio (2009)) and is the common practice at major BSSs such as BIXI in Montreal, Citi Bike in New York City and Ecobici in Mexico City. Such rebalancing is also a less expensive solution compared to installing more stations or to adding more docks to the current stations (Shu et al., 2013).

In most dock-based BSSs with operator-based rebalancing, the decision to actively rebalance a station depends on which stations are considered *unbalanced*. Depending on the BSS, the criteria may be different for a station to be categorized as such. For example, at NiceRide (Minneapolis, U.S)[1], a station is considered to be unbalanced when it either completely empty or completely full (Wang et al., 2018). The operators of Vélo'v (Lyon, France) [2] classify a station as unbalanced if the absolute difference between the number of arrivals and departures is larger than the standard deviation of the distribution of these values over all the stations (Borgnat et al., 2011). BIXI Montréal uses *inventory intervals* that establish an acceptable quantity of bikes at each station. Inventory intervals are manually set by BIXI's dispatching team, based on their experience on the station location, intraday demand fluctuation and day of the week.

[1] https://niceridemn.com/
[2] https://velov.grandlyon.com/

Nonetheless, the rebalancing process itself remains costly, as it accounts for gas, the maintenance of the truck fleet, drivers salaries, etc. In addition, it reduces the favourable impact that bike sharing claims to have on the environment. All considered, having a truck fleet large enough to rebalance all unbalanced stations every hour is not financially viable for most BSSs, especially during peak hours, and would erase most of the system's initial value proposition. According to JCDecaux, a company that offers self-service bikes to different cities around the world, the estimated cost in 2009 to relocate a single bike within a BSSs was about three dollars (DeMaio, 2009). Fleet trucks available for rebalancing are therefore limited in size and cannot rebalance all unbalanced stations. It becomes immediate that the planning of the rebalancing operations must be as effective as possible in order to satisfying the highest possible demand with the available resources.

Typically, multiple rebalancing operations are required every day in a BSS. Whenever the number of unbalanced stations exceeds the maximum rebalancing capacity of the system, the operator must select a subset of these stations to be rebalanced. Ideally, the subset of unbalanced stations should be selected such that the number of served future demand requests is maximized, which requires an appropriate demand forecast. However, predicting the demand of a BSS is a complex task depending on several factors, such as the weather, the hour of the day, the day of the week, holidays, public events, etc.

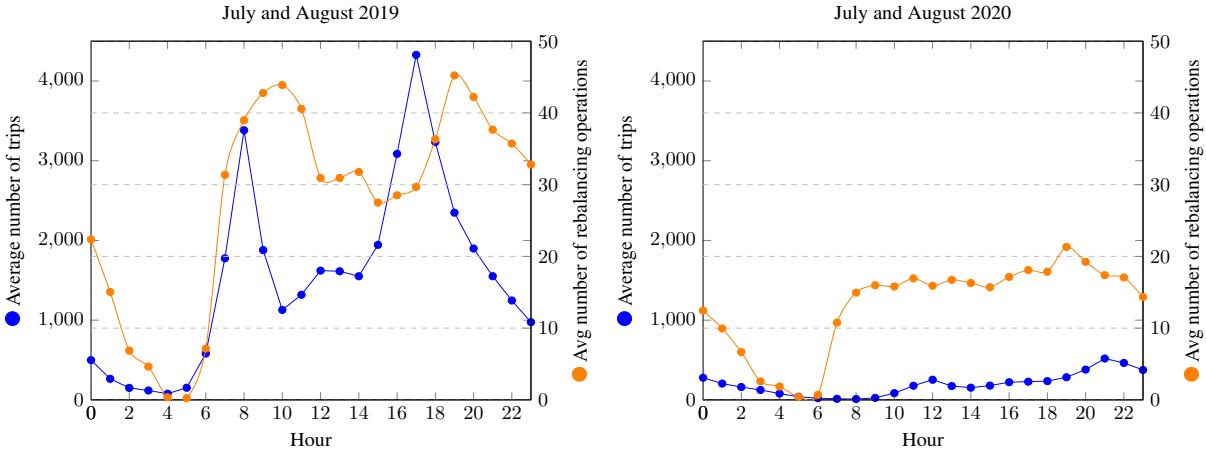


Fig. 1: Average number of trips and rebalancing operations for all stations per weekday.

Demand prediction in BSS became even more challenging in 2020 due to restrictive measures imposed by the governments in response to the COVID-19 pandemic, leading to a large part of the population working from home. Figure 1 shows the average number of trips and the average number of rebalancing operations performed by BIXI during the weekdays in July and August 2019 (left) and 2020 (right). Not only did the number of trips in 2020 decrease considerably with respect to the same period in 2019, but the trip behaviour also changed. In 2019, the peak hours happened right before and right after the working hours, i.e., at 8 a.m. and 5 p.m, respectively.

However, this pattern was no longer observed in 2020, resulting in a flatter demand along the day.

Such drastic demand changes as observed from 2019 to 2020 severely affect the subset of stations that become unbalanced over time, complicate the demand forecast and, in turn, affects the choice of stations to be rebalanced. As a result, the pattern of BIXI's rebalancing operations changed drastically, reducing their activities by nearly half during the peak hours in 2020. Given that the manual planning is mostly based on the previous experience of the dispatching team gathered, there is a significant risk that manually adjusted rebalancing strategies are ineffective in practice in a different environment of trip demand. This emphasizes the benefit of data-driven strategies to assist BSS rebalancing in order to quickly react to changing demand.

The main contribution of this paper is to highlight the benefits of using data-driven methods based on machine-learning in order to improve the performance of BSSs, thereby maximizing user satisfaction while minimizing service costs. To this end, we propose three strategies to select the stations that should be prioritized during the rebalancing process. These strategies are based on the current inventory levels at the stations, as well as the predicted demand for the next hours. The third strategy additionally considers the proximity among the unbalanced stations in order to prioritize stations that are close to each other. This may lead to smaller operational routing costs during an *a posteriori* truck route optimization.

Our three strategies are compared to a systematic approach, which is estimated to reproduce the prioritization strategy currently employed at BIXI. More specifically, the comparison is conducted by a tailored discrete-time simulation that computes the estimated lost demand (i.e., demand that could not be satisfied), the total number of alerts raised each time a station becomes unbalanced, and the number of performed rebalancing operations in the system. Within our computational experiments, the second proposed strategy reduced the estimated lost demand by 18% for the 2019 data and by 22% for the 2020 data, as compared to a baseline strategy that reproduces the prioritization performed by our BSS use case. The third prioritization strategy has demonstrated to select stations more naturally grouped while keeping good performance measure values. All proposed strategies can be easily implemented and are computationally cheap, therefore providing an attractive alternative to planning models based on mathematical optimization.

The remainder of this paper is organized as follows. Section 2 reviews the most relevant literature in the area of rebalancing and prioritization strategies for bike-sharing systems. Section 3 describes how the inventory intervals are defined so as to serve as input to the prioritization algorithms. Section 4 discusses the performance measures used to evaluate the various rebalancing strategies, as well as the simulator used to estimate such measures. Section 5 describes the different strategies propose to score the rebalancing priorities of unbalanced stations. Section 6 presents and analyzes the computational experiments. Finally, Section 7 concludes the paper.

2. Literature Review

Rebalancing in BSSs can be divided into two main steps: (a) inventory management, and (b) operational bike repositioning. Step (a) aims to set the number of bikes in each station to meet the

predicted demand as best as possible. Step (b) focuses on the actual dispatching operations that are necessary to achieve the desired station inventory levels in order to rebalance the system.

In order to define, in step (a), the optimal inventories that are likely to provide sufficient bikes and free docks to satisfy future demand, it is necessary to forecast future demand sufficiently well. Trip demand is influenced by numerous external factors, such as the weather, the day of the week, the time of the day, land use, the location of the stations, points of interest, and social-demographic characteristics (El-Assi et al., 2017; Hampshire and Marla, 2012). Most of the proposed approaches to predict trip demand are either based on machine learning (e.g. Feng et al. (2018); Hulot et al. (2018); Yin et al. (2012)) or on statistical models (e.g. Borgnat et al. (2011); Chen et al. (2016); El-Assi et al. (2017); Gebhart and Noland (2014)). These models differ from each other in terms of the predicted time horizon (hourly, daily or weekly), as well as the geographic granularity of the predictions (station-level, cluster-level or network-level).

For instance, Yin et al. (2012) and Gebhart and Noland (2014) predict the total demand in the network for each observed hour. Feng et al. (2018) and Borgnat et al. (2011) predict the total demand on a cluster-level, while Chen et al. (2016) estimate the probability that a cluster of stations becomes either completely full or empty. El-Assi et al. (2017) propose a model that estimates the future demand for each station for five periods of time along the day (morning, midday, afternoon, evening and overnight). Hulot et al. (2018) predict the hourly rentals and returns for each station, using temporal (day, day of the week, holiday, etc.) and weather (temperature, humidity, rain, etc.) features. The authors also propose a reduction technique to the trip data, improving the computational execution time and erasing outliers from the dataset. Once the demand is properly predicted, optimal inventories values can be determined. Schuijbroek et al. (2017) model the station inventory by means of a Poisson queuing system that estimates its optimal number of bikes while ensuring a given service level. In the work of Liu et al. (2016), the demand is predicted using a k-nearest neighbors regression model in which dissimilarities are weighted by meteorological forecast features. From the predicted number of trips, target inventories are optimized in order to maximize the amount of time a station is considered balanced. For practical purposes, station-level demand predictions for shorter time-periods (such as one hour) seems preferable given that (i) the demand can drastically change from one hour to the next, and that (ii) the rebalancing process is actually planned and carried out at station-level.

Regarding the operational decision-making step (b), the works in the literature can be categorized into two classes: those that assume that rebalancing operations are performed by the users of the system (under some incentive) and those that rebalance by means of a truck fleet controlled by the operator. Chemla et al. (2013b) propose a reward mechanism to encourage users to return bikes to certain stations in the BSS. In Fricker and Gast (2016), the authors conclude that encouraging the users to return the bikes to a non-saturated station does not significantly improve the system's performance. However, they also show that the performance can be improved, by constantly stimulating users to return bikes to a nearby station with lower inventory.

In the case of dock-based BSS, as the one approached here, operator-based rebalancing via trucks has typically been modelled via mixed-integer linear programming (e.g. Alvarez-Valdes et al. (2016); Brinkmann et al. (2016); Bulhões et al. (2018); Chemla et al. (2013a); Contardo et al. (2012); Dell'Amico et al. (2014); Erdoğan et al. (2015); Lowalekar et al. (2017); Pal and Zhang (2017); Papazek et al. (2013)). These models generally aim at finding optimal truck routes

to rebalance a set of stations, typically aiming to maximize customer satisfaction. The latter may be achieved by minimizing the total lost demand (see, e.g. Alvarez-Valdes et al. (2016); Contardo et al. (2012); Lowalekar et al. (2017)), keeping station inventories close to their respective target inventories (see, e.g. Brinkmann et al. (2016); Papazek et al. (2013)) or even by optimizing several (possibly conflicting) objectives (see, e.g. Nunes et al. (2022)). Unfortunately, the use of such models in practice is rather challenging, given that the resulting optimization models tend to be hard to solve. This typically limits their use to a small number of stations, given that an intraday planning typically requires decisions within a matter of minutes. Schuijbroek et al. (2017) observe that their formulation becomes difficult to solve even for a small instances with 50 stations and 3 trucks. The authors therefore propose a heuristic that clusters stations using a maximum spanning star and then rebalance among clusters. Likewise, Ghosh et al. (2017) cluster nearby stations, and then rebalance among the clusters, where the capacity and the inventory of each cluster is given by the sum of its stations. While such an approach is improves computational feasibility, it is based on the assumption that nearby stations have similar patterns and that rebalancing within each cluster is time feasible.

Several other heuristic methods have been proposed (e.g. Lu et al. (2020); Papazek et al. (2014); Ren et al. (2020)). In particular, Vergeylen et al. (2020) propose a large neighbourhood search algorithm that optimizes the truck routes only for stations that have raised an alert to the system. An interesting characteristic of their optimization model is that such alerts have different priorities which are proportional to their importance in the objective function. The list of stations to rebalance (i.e., those that raised an alert) along with their associated priorities have to be provided as input to the optimization model.

Such approaches, based on alerts raised for stations that are prone to become unbalanced, are also easier to fit to existing practices at several BSSs, who often plan the dispatching operations based on such alerts. Nonetheless, because of limited resources in practice, planners are often required to choose a subset of the unbalanced stations. Indeed, a prior selection of stations can be very useful to scale optimization models to large BSSs by restricting the amount of stations to be actually considered.

The algorithms proposed in the next sections seek to recommend the best subset of stations to be rebalanced over a prespecified period of time (e.g. one hour). Given that these prioritization strategies are easy to implement and computed within a matter of seconds, they provide an attractive alternative to computationally expensive optimization models and can easily complement systems that perform rebalancing planning based on raised station alerts.

3. Inventory intervals and service levels

Keeping station inventories at specific target values is practically infeasible. BSS operators (such as BIXI Montreal) therefore often use intervals of acceptable inventory levels, referred to as *inventory intervals*. They are composed of a lower and an upper bound, as well as the target value of the interval itself, which refers to the ideal inventory for that station and is located within the lower and the upper bound. Typically, each station has its specific inventory interval defined for a specific time period, and its values may change depending on hour and day. When the inventory of a station

falls outside of its specified inventory interval, the station is classified as unbalanced, which in turn triggers a rebalancing alert.

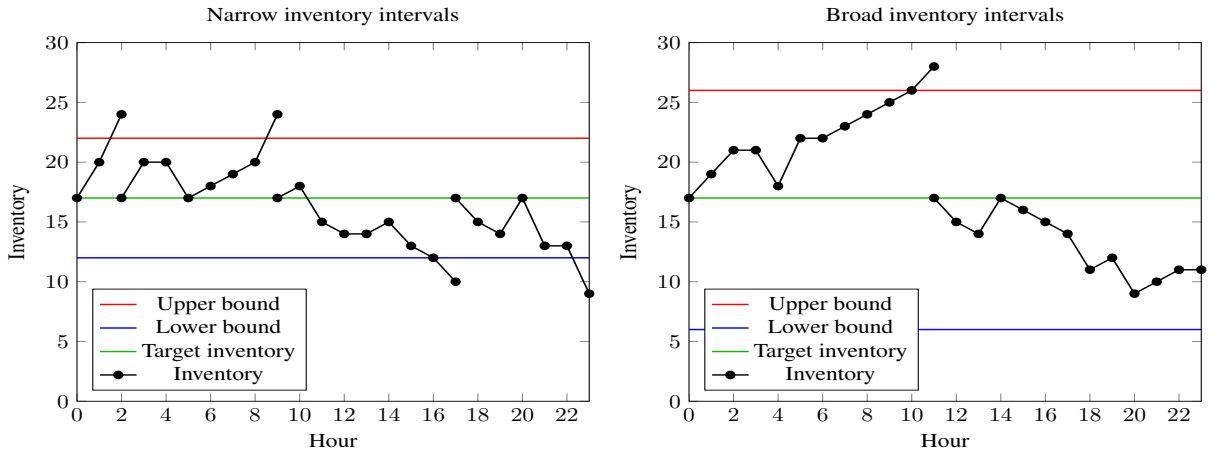


Fig. 2: Trade-off between the number of alerts and lost demand when defining inventory intervals.

Inventory intervals have a major impact on the quantity of lost demand, on the number of raised alerts and ultimately, on the total number of rebalancing operations. For example, narrow intervals generate more alerts, but tend to keep the inventory closer to its target value, which in turn, decreases the likelihood of lost demand (see Figure 2, left). In contrast, wide inventory intervals create fewer alerts, but at the expense of keeping the inventory farther from its target values, which may increase the number of lost demand (see Figure 2, right). Defining inventory intervals that make this delicate trade-off in order to maximize system performance over the entire day is therefore a sensitive challenge.

Hulot et al. (2018) propose a model that automates the creation of inventory intervals using demand prediction based on historical trips, weather and temporal data. The model first predicts future demand and then computes the appropriate service levels that translate into inventory intervals. The first part aims at predicting as accurately as possible the hourly rental and return demand for each station. Among the tested machine learning models, the best regression performance was obtained by a gradient boosted tree (GBT). As our objective is to provide a prioritization ranking among all stations throughout the day, we adapt the GBT model in order to predict future demand. As such, the prioritization strategies presented in this paper (see Section 5) take the station-level demand prediction as an input. In the following, we will briefly review the concepts proposed by the predictive model.

After forecasting the demand for each station, the model computes its *service level*, which is defined as the ratio between the estimated number of realized trips and the total trip demand at that station. The inventory of the stations is modelled as an M/M/1/K queue, whose parameters are the time between the rentals, the time between the returns, the number of servers (here, a single station) and the maximum capacity of the server (i.e., the total number of docks C_s). Since the

authors assume that the trips are described by a Poisson distribution, the times between rentals and returns follow exponential distributions. Thus, given a station s with initial inventory f, a time period T, and the parameters mentioned above, the rental and the return service levels are computed as:

$$SL_{rental}(f, s, T) = \frac{\int_0^T \mu_s(t)(1 - p_s(f, 0, t))dt}{\int_0^T \mu_s(t)dt} \tag{1}$$

$$SL_{return}(f, s, T) = \frac{\int_0^T \lambda_s(t)(1 - p_s(f, C_s, t))dt}{\int_0^T \lambda_s(t)dt}, \tag{2}$$

where $p_s(f, N, t)$ is the probability that the station s has N bikes at time t, assuming that it had f bikes at time 0, and $\mu_s(t)$ (resp. $\lambda_s(t)$) is the expected demand at time t at station s for renting (resp. returning) bikes. The conservative overall service level is then defined as:

$$SL(f, s, T) = \min\{SL_{rental}(f, s, T), SL_{return}(f, s, T)\}). \tag{3}$$

The authors of Hulot et al. (2018) also introduce a hyperparameter α to prioritize either rentals or returns. This hyperparameter increases the flexibility of the model by adjusting service level computation to periods of the day during which the rental demand is higher than the return demand, and vice-versa. The overall service level parametrized by α is then computed by:

$$SL_\alpha(f, s, T) = \min\{\alpha SL_{rental}(f, s, T), (1 - \alpha)SL_{return}(f, s, T)\}. \tag{4}$$

Thus, $\alpha > 0.5$ emphasizes bike returns, while $\alpha < 0.5$ emphasizes rentals.

We observe that the minimum and maximum service levels for a station s in a time period T depend on the initial inventory at time 0, and are given by:

$$SL_{min}(s, T) = \min_{f \in \{0,\dots,C_s\}} (SL_\alpha(f, s, T)), \tag{5}$$

and

$$SL_{max}(s, T) = \max_{f \in \{0,\dots,C_s\}} (SL_\alpha(f, s, T)). \tag{6}$$

In order to establish a threshold Ω for the acceptable service level for a station s in time period T, we compute:

$$\Omega(s, T) = SL_{min}(s, T) + \beta(SL_{max}(s, T) - SL_{min}(s, T)), \tag{7}$$

where the hyperparameter β controls how exigent the operator is about the network. A small value of β approximates the threshold $\Omega(s, T)$ to the minimum service level, while a large β brings the

threshold $c(s,T)$ closer to the maximum service level.

The inventory interval for station s for time period T is then calculated as follows:

$$\mathcal{I}(s,T) = \{f \in \{0,...,C_s\} | SL_\alpha(f,s,T) \geq \Omega(s,T)\} \tag{8}$$

Finally, the target value for a station s for a time period T is equal to the inventory value $f \in \mathcal{I}(s,T)$ for which $SL_\alpha(f,s,T)$ is largest.

4. Measuring BSS performance via simulation

Operators of dock-based BSSs evaluate the performance of their rebalancing operations mainly using three measures: the number of raised alerts, the number of rebalancing operations carried out and the ability of satisfying rental and return demand. The number of raised alerts indicates how often stations have been classified as unbalanced. The number of rebalancing operations typically depends on the raised alerts and impacts the operating costs. Finally, the results of the previous two elements is the number of rental and return requests that could not be satisfied, also referred to as *lost demand*, which directly affects customer satisfaction. In this section, we will explore how we may estimate these three measures in order to classify the performance of a rebalancing strategy, which in this paper, is given by a systematic strategy to prioritize stations. To this end, we first review in Section 4.1 the inventory simulator propose by Hulot et al. (2018). Given that this simulator assumes the worst-case inventory given by the lower and upper bound of the stations inventory intervals, its outcome is useful to rank different rebalancing strategies, but the estimated performance measures are likely to be less accurate. We therefore propose a more realistic simulator in Section 4.2, which simultaneously computes the number of alerts, demand losses and rebalancing operations. This enables us to update the estimated inventory intervals depending on the prioritization strategy for the rebalancing operations in a rolling-horizon fashion and obtain a more realistic estimate of the three aforementioned performance measures.

4.1. Inventory simulator based on worst-case inventory levels

Hulot et al. (2018) propose two distinct ways to estimate the number of alerts raised by an BSS as well as the lost demand of trips. To compute the first, their estimation assumes that the bike inventories of all stations start at their target values, and that the operator has unlimited rebalancing capacity. Then, at each hour, the inventory of the stations is updated according to the amount of rented and returned bikes. Whenever the inventory of a station lies outside its inventory interval, an alert is generated and a truck is dispatched to restore the inventory of that station to its target value for the next hour.

Computing lost demand is, however, more challenging because it is usually related to unobserved data (Tan and Karabati, 2004). That is, there is no information about how many commuters wanted to rent or return a bike at a specific station, but could not due to starvation (i.e, no bikes available) or due to congestion (i.e., no empty docks available). Besides, it is common for the com-

muters to substitute an unavailable station by others, masking its lost demand and increasing the demand in its nearby stations. Hulot et al. (2018) propose to estimate the lost demand considering a worst-case scenario which computes the lost demand considering that the inventory of a station is always at its maximum value for returns, and at its minimum value for rentals. For example, assume that the interval's lower bound for a given station is 3 and that 5 commuters want to rent a bike at a given hour. The estimated lost rental demand is then computed as $\max\{0, 5-3\} = 2$. Likewise, if we assume an upper bound of 12 for a station having 15 docks, and that 6 commuters want to return a bike, the estimated lost return demand is computed as $\max\{0, 6-(15-12)\} = 3$.

While the outcome of such a simulation may be useful to provide an approximate ranking among different rebalancing strategies, in practice, the assumption of station inventories at their interval bounds is rather conservative and may rarely occur. We propose next a more realistic inventory simulation that takes into consideration the actual operator's rebalancing capacity, as well as the estimated inventory level at each station.

4.2. Inventory simulator based on expected inventory level

In view of the shortcomings exposed above, we developed a new simulation tool to compute the number of raised alerts, lost demand and rebalancing operations in simulated BSSs. Our simulation takes into account the constraints of the bike-sharing systems regarding the number of stations that can be rebalanced throughout each time period (in our case, each hour). As a consequence, at each time period, only a subset of unbalanced stations can be rebalanced and have their inventory intervals set back to their respective target value.

Algorithm 1 presents the pseudo-code of the simulator. The algorithm receives input matrices containing the inventory intervals' lower bounds ($lower$), upper bounds ($upper$) and target values ($target$) for all stations and all simulated time periods. These matrices are pre-computed as by equation (8). Further input parameters include the demand matrices for $rentals$ and $returns$ at each station and for each time period, the vector of available docks at each station ($\#docks$), and a variable representing the operator's rebalancing capacity ($rebalancing\_capacity$), referring to the number of stations that can be rebalanced.

The simulator first initializes the station inventories to their respective target value. At each subsequent time period, their inventories are updated according to the expected rental and return demand. The simulation starts with a loop in lines 1-7 that iterates over the stations to initialize the current inventories ($inventory$) and the accumulator variables for the number of alerts ($alerts\_returns$ and $alerts\_rentals$), and lost demand ($alerts\_rentals$ and $lost\_returns$). As well, in line 8, the variable $rebalancing\_ops$, that stores the number of rebalancing operations done so far in the simulated time period, is initialized. Next, the algorithm iterates over the simulated hours of the considered time period in lines 9-33. It assumes an empty set of $unbalanced$ stations at the beginning of each simulated hour. In the loop of lines 11-25, the algorithm proceeds by updating the inventory of each station s in line 12. If the inventory of s is above its upper bound in line 13 a return alert is raised. On the contrary, if the inventory of s is below its lower bound in line 16 a rental alert is raised. In both cases, the station s is added to the set of unbalanced stations. Likewise, lost demand is computed in lines 20-24 if the inventory of s lies below 0 or above its

number of docks.

After iterating over all the stations, the algorithm calls the function PRIORITIZE in line 26 that sorts all the unbalanced stations according to their priority. Then, in lines 27-31, the inventory of the prioritized stations are rebalanced to their target values up to the system's rebalancing capacity. In the sequel, line 32 updates the number of performed rebalancing operations during the simulation. Finally, the algorithm returns in line 33 information about the number of raised alerts, lost demand and total rebalancing operations, which are used to assess the performance of the system.

5. Prioritizing strategies for unbalanced stations

As discussed in the previous sections, the rebalancing capacity available to the system operator may not be sufficient to rebalance all stations that have raised an alert. Figure 3 shows BIXI's estimated hourly rebalancing capacity, as well as the average daily number of rental and return alerts raised at BIXI's stations, at each hour, for weekdays in July and August 2019 (left) and 2020 (right). Here, the number of rental and return alerts have, most of the time, been significantly superior to the BIXI's rebalancing capacity (about 46 stations in 2019 and about 22 stations in 2020).

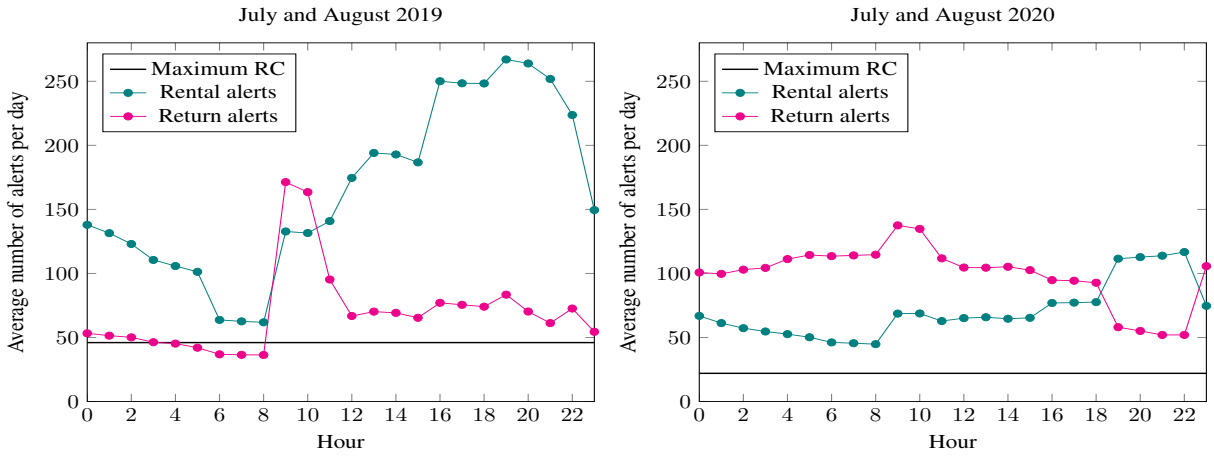


Fig. 3: Maximum rebalancing capacity and average daily number of alerts raised by BIXI's stations at each hour during weekdays in July and August 2019 and 2020.

Operators typically follow a systematic approach how to prioritize stations for rebalancing among those that have raised an alert, which we here refer to as *rebalancing strategies*. In the following, we will first review the prioritization strategy currently in place at BIXI in Section 5.1. In the remaining three sections, we then propose three data-driven prioritization strategies, which explicitly consider data such as the current and future demand, station inventory and location.

Algorithm 1 Simulation algorithm to compute alerts and lost demand

Input: $target$,$lower$,$upper$,$returns$,$rentals$,$\#docks$,$rebalancing\_capacity$

for each station s **do**
  $inventory[s] \leftarrow target[s, 0]$
  $alerts\_returns[s] \leftarrow 0$
  $alerts\_rentals[s] \leftarrow 0$
  $lost\_returns[s] \leftarrow 0$
  $lost\_rentals[s] \leftarrow 0$
end for
$rebalacing\_ops \leftarrow 0$
for each simulated hour h **do**
  $unbalanced \leftarrow \emptyset$
  for each station s **do**
    $inventory[s] \leftarrow inventory[s] + returns[s, h] - rentals[s, h]$
    if $inventory[s] > upper[s, h]$ **then**
      $alerts\_return[s] \leftarrow alerts\_returns[s]$ + 1
      $unbalanced \leftarrow unbalanced \cup \{s\}$
    else if $inventory[s] < lower[s, h]$ **then**
      $alerts\_rentals[s] \leftarrow alerts\_rentals[s]$ + 1
      $unbalanced \leftarrow unbalanced \cup \{s\}$
    end if
    if $inventory[s] > \#docks[s]$ **then**
      $lost\_returns[s] \leftarrow lost\_returns[s] + (inventory[s] - \#docks[s])$
    else if $inventory[s] < 0$ **then**
      $lost\_rentals[s] \leftarrow lost\_rentals[s] - inventory[s]$
    end if
  end for
  $prioritized\_stations \leftarrow$ PRIORITIZE($unbalanced$,$inventory$,$lower$,$upper$,h)
  $\#rebalanced \leftarrow 0$
  while $\#rebalanced < \min\{|prioritized\_stations|, rebalancing\_capacity\}$ **do**
    $inventory[prioritized\_stations[\#rebalanced]] \leftarrow target[prioritized\_stations[\#rebalanced], h]$
    $\#rebalanced \leftarrow \#rebalanced + 1$
  end while
  $rebalacing\_ops \leftarrow rebalacing\_ops + \#rebalanced$
end for
return $alerts\_returns$, $alerts\_rentals$, $lost\_returns$, $lost\_rentals$, $rebalancing\_ops$

5.1. BIXI's prioritization strategy

BIXI uses a systematic approach to select the subset of unbalanced stations that should be prioritized for rebalancing operations. Their approach is summarized by the flowchart[3] in Figure 4. The

[3]The presented flowchart was obtained after several exchanges with BIXI's planners. As such, it is not an official representation of BIXI's decision-making process.

approach can be divided into three parts. First, stations are classified as *critical*, if they are either completely empty (or completely full) and for which all neighbouring stations within a radius of 600 meters are also completely empty (or completely full)[4]. In this case, a lack of available bikes or empty docks at the current or surrounding stations would result in lost demand. If the number of critical stations is larger than (or equal to) the maximum rebalancing capacity N, the N critical stations that are the closest to a metro station are rebalanced [5]. If the number of such stations is larger than the remaining rebalancing capacity, the second part of the approach rebalances stations that are considered non-critical, but are close to a metro station (specifically, within a reach of 600 meters). If the remaining rebalancing capacity is smaller than the number of such stations, those that are closest to a metro station will be prioritized. Otherwise, all of those stations will be rebalanced and the rebalancing capacity remaining thereafter will be allocated in the third part. This final part considers all remaining unbalanced stations that are within a reach of 600 meters from any stations that have already been selected for rebalancing. This condition aims at leveraging trucks that will already be sent close to such stations. Among those, stations are prioritized that have the most unbalanced inventory.

Note that this prioritization strategy may disregard a certain amount of unbalanced stations (i.e. stations that raised an alert) in case they do not meet the established conditions.

5.2. Prioritization strategy based on inventory forecasting

The first prioritization algorithm we propose, denoted Pa_1, selects stations for rebalancing taking into consideration the current station inventories, provided as input, and the expected demand, which is obtained by the regression model of Hulot et al. (2018). We predict the inventory of each station s in the network for the next hour $h+1$ as follows:

$$pred\_inventory[s, h+1] = inventory[s] + pred\_returns[s, h] - pred\_rentals[s, h], \quad (9)$$

where $pred\_rentals[s, h]$ and $pred\_returns[s, h]$ are, respectively, the expected demand for the number of rentals and returns at s during the simulated hour h, and $inventory[s]$ refers to the current number of bikes available at s.

Note that (9) can also be extended to predict inventories for subsequent hours in advance as well. For that, it suffices to predict bike demands for the hours of the considered time period.

Then, for each station s in the system, the algorithm computes two indices

$$bikes^{-}[s] = \max\{0, -pred\_inventory[s, h+1]\} \quad (10)$$

[4]The radius of 600 meters is defined by BIXI based on the fact that an average person may walk this distance within 10-15 minutes, which is the time that BIXI considers acceptable for a commuter to walk seeking to be served.
[5]If the number of critical stations equals the (remaining) rebalancing capacity, all stations are rebalanced without sorting.

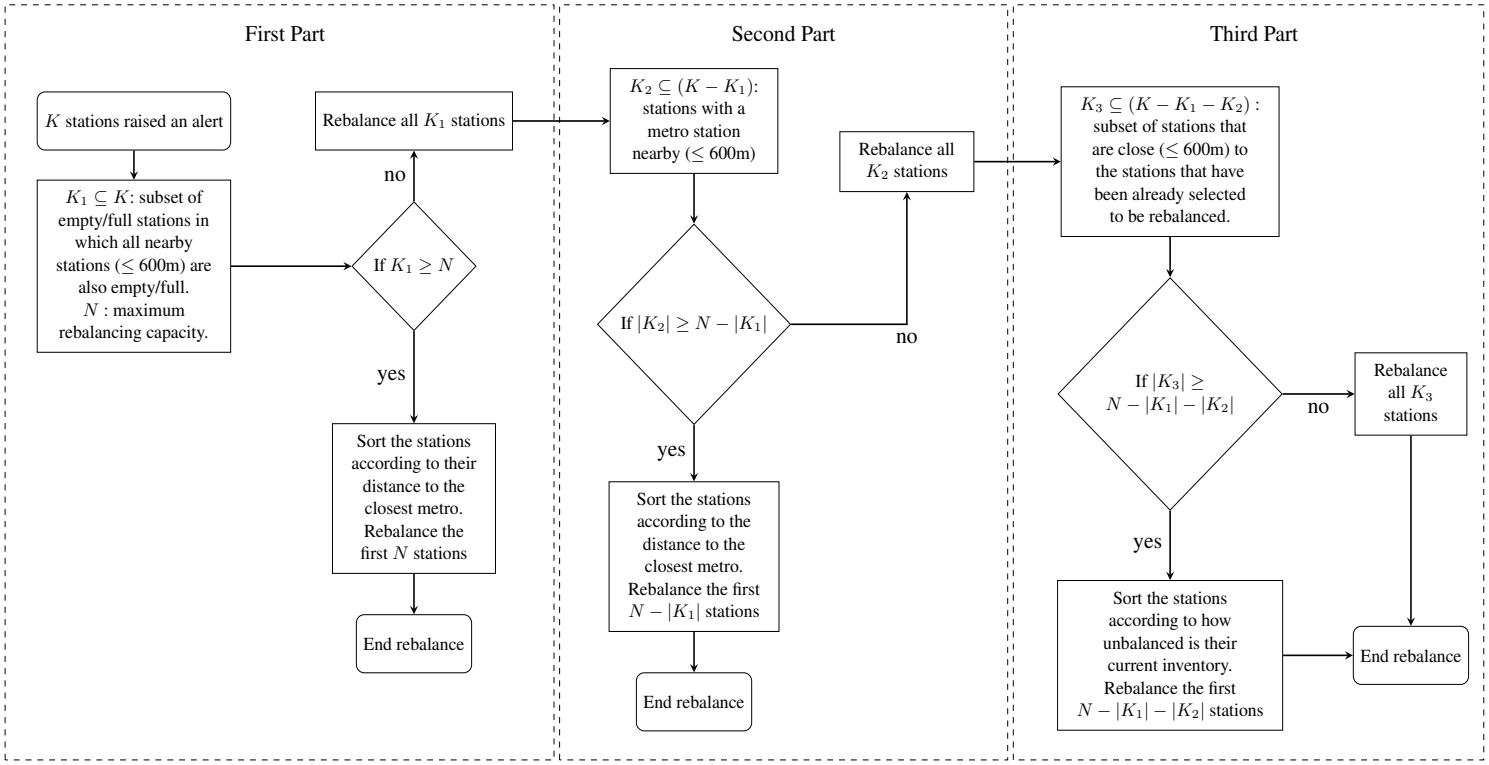


Fig. 4: Flowchart of BIXI's prioritization strategy.

and

$$bikes^{+}[s] = \max\{0, pred\_inventory[s, h+1] - \#docks[s]\}, \tag{11}$$

which represent the predicted inventory shortfall and surplus, respectively. By definition, only one of the two indices $bikes^{-}[s]$ and $bikes^{+}[s]$ can assume a value larger than 0. Finally, a prioritization score, denoted $priority_1$, is computed as:

$$priority_1[s] = \max\{bikes^{-}[s], bikes^{+}[s]\}. \tag{12}$$

Hence, strategy Pa_1 prioritizes stations for which a loss of demand is predicted for the next hour(s). The stations are therefore prioritized in non-increasing order according to their predicted inventory shortfall or surplus.

5.3. Prioritization strategy using inventory intervals

The second prioritization algorithm we propose, denoted Pa_2, is less conservative than Pa_1. Instead of giving a high prioritization score to stations whose inventories are predicted below 0 or above its dock capacity, this strategy prioritizes according to the deviation of the station's inventory from its inventory interval, as defined by (8). Therefore, indices (10) and (11) are replaced by:

$$bikes^{-}_{\mathcal{I}}[s] = max\{0, lower[s,h] - pred\_inventory[s,h+1]\} \tag{13}$$

and

$$bikes^{+}_{\mathcal{I}}[s] = max\{0, pred\_inventory[s,h+1] - upper[s,h]\}. \tag{14}$$

The final priority score provided by Pa_2 is then computed as:

$$priority_2[s] = \max\{bikes^{-}_{\mathcal{I}}[s], bikes^{+}_{\mathcal{I}}[s]\}. \tag{15}$$

As a result, Pa_2 prioritizes stations that are expected to raise an alert within the next hour. These stations are therefore sorted in non-increasing order according to how much their predicted inventories deviate from their inventory interval bounds.

5.4. Prioritization strategy based on neighbourhood

Our third prioritization algorithm, denoted Pa_3, proposes a modification to the priority score computed by Pa_2 and favours the rebalancing of neighbouring stations. Thus, Pa_3 is expected to result in geographically more compact and thus less costly (and faster) dispatching routes, grouping together nearby stations.

Let us define $\mathcal{N}^s$ as the index set of all stations within a radius of 600 meters from station s. The priority score computed by Pa_3, denoted $priority_3$, is then given by:

$$priority_3[s] = \begin{cases} \gamma \times priority_2[s] + (1-\gamma) \times \frac{\sum_{s' \in \mathcal{N}^s, s' \neq s} priority_2[s']}{|\mathcal{N}^s|}, & \text{if } priority_2[s] > 0 \\ priority_2[s], & \text{otherwise.} \end{cases} \tag{16}$$

The hyperparameter γ controls the weight of the neighbourhood inventory information on the computation of the priority score. By using $\gamma < 1$, the priority score computed by Pa_3 for a station s takes also into consideration the priority scores of its neighbouring stations. As γ approaches 0, those scores tend to prevail in (16), prioritizing stations surrounded by others that raised an alert to the system.

In all the proposed prioritization algorithms, stations are sorted in non-increasing order of their computed priority scores, and are then returned by function PRIORITIZE (see pseudo-code of Al-

gorithm 1). Stations with a priority score of 0 are simply discarded and, in the case where stations have the same score, they are sorted based on their proximity to their closest metro station.

6. Computational experiments

We now report on computational experiments that were carried out to evaluate the performance of the various prioritization strategies. We first present the details of the dataset used in our study. In the sequel, we discuss the selection of the hyperparameters α and β, used to define inventory intervals, as well as γ, used by prioritization algorithm Pa_3. We then compare the estimated system performance induced by the different prioritization algorithms by means of lost demand, the number of raised alerts and the number of performed rebalancing operations. Finally, we assess Pa_3 with respect to the clustering property of its prioritized stations.

6.1. Data set

The dataset used in the experiments contains hourly information regarding time, weather, trips and stations. The time features hold temporal data such as the hour, day, holiday and the day of the week. The weather features store information such as temperature, wind speed, relative humidity, etc. The trip features are composed by the number of bike rentals and returns observed at each station of the network. At last, the stations features contain information regarding the geographical location of each station, as well as their corresponding number of docks. Time and weather features were collected from `https://climate.weather.gc.ca` (except for the holiday feature that was manually collected), while trip and station features were both provided by BIXI-Montreal. More details about the importance of the different features for the GBT used here can be found in Hulot (2018).

The dataset was split between training, validation, and test datasets. The training data was used to fit the machine learning model parameters. The validation dataset was used to tune the hyperparameters used by the prioritization algorithms. Finally, the test dataset was used to provide an unbiased performance evaluation of the different prioritization strategies.

Because of a large observed discrepancy in the frequency and the behaviour of trips in 2020, caused by the COVID-19 pandemic, we used different strategies for selecting the dataset for each simulated season. For the 2019 tests, the training dataset contains data from April 2018 to June 2019, minus the months during which BIXI is out of service (i.e., December, January, February and March). The validation dataset is composed of data from the first halves of July and August 2019, and the test dataset uses data from the second halves of July and August 2019. We opted to divide both July and August into validation and test datasets so that the model is less sensitive to demand changes observed between consecutive months.

Note that the physical network of BSSs typically changes over time, which makes it often difficult to use data linked to a specific station ID over longer periods of time. We therefore do not use data from 2017 and before, since a large number of BIXI stations changed their location and/or their capacity without changing their IDs in the provided databases after 2017. We have also ob-

served that, for the 2020 season, adding training data from previous years deteriorates the traffic prediction. Consequently, we used training data from April 2020 to June 2020. In that case, the validation dataset contains the first halves of July and August 2020, while the test dataset contains the second halves of July and August 2020. Note that we focus our experiments on the months of July and August because of their high demand and importance throughout the season.

Figure 5 depicts our simulation pipeline. For demand prediction, the algorithm uses trip, weather and temporal data in order to learn trips patterns so as to predict for the next hours the number of bike rentals and returns at each station. The predicted demand is then used along with the station data (i.e., number of docks per station) to generate the inventory intervals for each of the stations. Finally, the simulator estimates the inventory and the performance measures based on: (i) the station and trip data (seen - training, or unseen - validation/test); (ii) the computed inventory intervals; (iii) the predicted future demand; (iv) the maximum rebalancing capacity; and (v) the ranked station lists from the prioritization strategies.

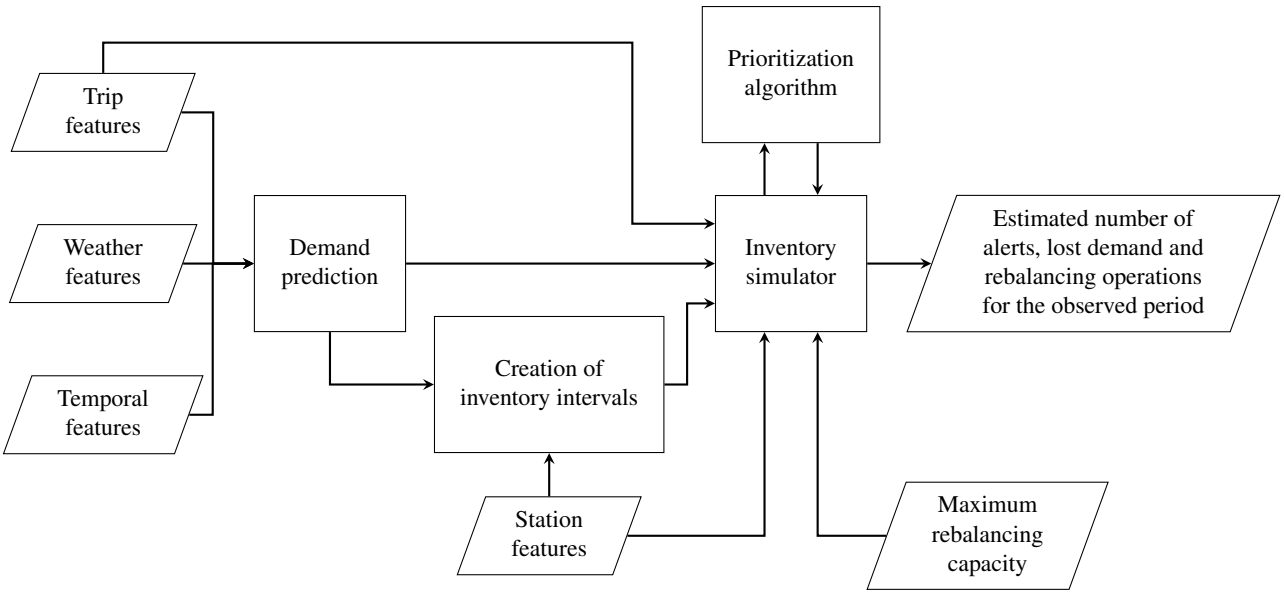


Fig. 5: Simulation pipeline.

6.2. Hyperparameter Tuning Process

In order to define inventory intervals that achieve a good performance during simulation, hyperparameters α and β (see Section 3) have to be tuned to the network. Ideally, the values of these hyperparameters should lead to inventory intervals that adapt to the expected upcoming demand, thus yielding small numbers of alerts, lost demand and rebalancing operations. However, the considered performance measures might be pairwise conflicting. For example, a small number of

alerts tend to trigger less rebalancing operations in the system, which in turn, may increase the lost demand.

With that in mind, we decided to set the values of α and β by examining their associated Pareto-frontier Rosenthal (1985), composed by the set of non-dominated pair of values regarding (i) the lost demand, (ii) the number of raised alerts, and (iii) the number of rebalancing operations objectives. A pair of values (α', β') is said to be dominated by another pair (α'', β'') if the latter is at least as good as the first regarding (i), (ii) and (iii), being strictly better for at least one of them. We approximated the Pareto-frontier using a grid-search in $\alpha, \beta \in [0.20, 0.25, \ldots, 0.80]$.

Hereafter, our results refer to three non-dominated Pareto solutions of interest, namely:

- Combination $A = (\alpha_A, \beta_A)$, which yielded the lowest volume of lost demand;
- Combination $B = (\alpha_B, \beta_B)$, which yielded the lowest amount of raised alerts and rebalancing operations; and
- Combination $C = (\alpha_C, \beta_C)$, which is associated to the median of the ordered set of solutions in the Pareto Frontier.

Table 1 reports the optimized hyperparameters values obtained in our computational tests. Particularly, we observe that the value of $\alpha = 0.5$ appears to be optimal in all cases, which means that the inventory intervals are defined such that they do neither prioritize rentals nor returns (see Section 3).

Table 1: Optimized hyperparameters values used in the tests.

Algorithm	Combination	2019 dataset	2020 dataset
BIXI	(α_A, β_A) (α_B, β_B) (α_C, β_C)	(0.50, 0.20)	(0.50, 0.20)
Pa_1	(α_A, β_A) (α_B, β_B) (α_C, β_C)	(0.50, 0.50) (0.50, 0.20) (0.50, 0.30)	(0.50, 0.80) (0.50, 0.20) (0.50, 0.50)
Pa_2	(α_A, β_A) (α_B, β_B) (α_C, β_C)	(0.50, 0.30) (0.50, 0.20) (0.50, 0.25)	(0.50, 0.40) (0.50, 0.20) (0.50, 0.30)

We also remark that the approximated Pareto frontiers obtained by the BIXI's prioritization algorithm for the 2019 and 2020 seasons have a unique solution. This is explained by the fact that BIXI's prioritization strategy prioritizes stations that are located close to a metro station (see Figure 4). Higher β values imply more narrow inventory intervals, leading to an increased number of stations that raise an alert. Therefore, stations located far from a metro station are less likely to be rebalanced. As a result, the (α, β) combination with lowest β value was found to be the one that performed best under BIXI's prioritization strategy.

Finally, we remark that the β values, for combinations A and C, are larger for 2020 data. As the trip demand was considerably smaller in 2020 than in 2019, the value of β is allowed to be further

Table 2: Collected results from 2019 and 2020 season.

Season	Algorithm	Combination	Lost Demand		Alerts		Rebalancing Operations	
			Value	Diff	Value	Diff	Value	Diff
2019	BIXI	*A, B and C*	29480		41230		25474	
	Pa_1	*A*	30798	4.47	100254	143.16	22764	-10.64
		B	31130	5.60	57553	39.59	22278	-12.55
		C	30963	5.03	69925	69.60	22547	-11.49
	Pa_2	*A*	24129	-18.15	43246	4.89	25446	-0.11
		B	24283	-17.62	36994	-10.27	23481	-7.82
		C	24185	-17.96	39738	-3.62	24441	-4.06
2020	BIXI	*A, B and C*	7490		13174		10890	
	Pa_1	*A*	9096	21.44	117431	791.38	9728	-10.67
		B	9171	22.44	30509	131.58	9625	-11.62
		C	9103	21.54	60377	358.30	9671	-11.19
	Pa_2	*A*	5859	-21.78	16676	26.58	12212	12.14
		B	6809	-9.10	12515	-5.00	10503	-3.56
		C	6118	-18.32	14104	7.06	11231	3.13

increased for reducing the lost demand before being limited by the system's rebalancing capacity.

While we have considered hyperparameters α and β above, note that the selection of hyperparameter γ (as used in (16)), will be examined later in section 6.3.1 along with strategy Pa_3.

6.3. Simulation results

We now report on the simulation results of BIXI's data for the 2019 and 2020 seasons using the hyperparameters selected above. Specifically, we compare the results of each prioritization algorithm regarding the number of raised alerts in the stations, the amount of lost demand and the number of performed rebalancing operations.

The capacities of maximum hourly rebalancing operations used in our simulator for the 2019 and 2020 seasons were set to 46 and 22, respectively, which corresponds to the average number of rebalancing operations performed by BIXI during the peak rebalancing hours in the two years.

Table 2 presents the number of alerts, lost demand and rebalancing operations estimated by the simulator (see Section 4.2) for each of the prioritization algorithms and hyperparameter combinations. We present the absolute results with the estimated value for each performance measure, as well as relative results (in %) with respect to the measure values obtained by the BIXI's prioritization strategy.

The trip demand for 2019 was considerably higher than in 2020 (also compare Figure 1), when the measures applied by the Canadian authorities in response to the COVID-19 pandemic encouraged the population to social distancing and working from home. As a consequence of these mea-

sures, both the total number of trips and the trip behaviour drastically changed. Since more data was used for training the trip prediction model in 2019, the results obtained by the proposed prioritization algorithms for that year are usually better than those obtained for 2020, when compared to the BIXI's prioritization strategy.

Moreover, we observe that Pa_1 is not effective as a prioritization algorithm. Recall that Pa_1 only prioritizes stations for which a shortfall or a surplus is predicted for the next hour (see Equations (10-12)), even if an alert is raised for those stations. This turns out to be a rather restrictive rebalancing criterion, resulting in a considerably smaller number of rebalancing operations using Pa_1 compared to the other prioritization algorithms, although the number of alerts is still large.

In contrast, strategy Pa_2 successfully reduces the lost demand, attaining a reduction of $\approx 22\%$ (using hyperparameter combination A for 2020) with respect to BIXI's prioritization strategy. Regarding the number of alerts and rebalancing operations, Pa_2 (with combination B) improves these performance measures in both 2019 and 2020 seasons, reaching a reduction of $\approx 10\%$ in the number of alerts and $\approx 8\%$ in the number of rebalances in 2019. In particular, Pa_2 with hyperparameter combination B improves all three measures for both seasons.

6.3.1. Evaluation of Pa_3

In this section, we turn our analysis to the prioritization algorithm Pa_3, which is designed to favour the rebalancing of proximal unbalanced stations as γ tends to zero. Thus, it is expected that less lengthy, and consequently, less expensive routes can be derived for the trucks that are in charge of the system rebalancing.

Let $G = (N, E)$ be a graph for which there is a node $n_i \in N$ corresponding to each station selected for rebalancing at a given hour. Set E contains all edges e_{ij} for which $i \in N$ and $j \in N$ and i and j are not more than 600m apart. Let us denote $S_i \subseteq N$ the set of nodes connected to $n_i \in N$.

In order to evaluate Pa_3 regarding its intended objective, we compute the *Watts–Strogatz clustering coefficient* of G given by:

$$\eta_G = \frac{1}{|N|} \sum_{n_i \in N} \eta_i, \tag{17}$$

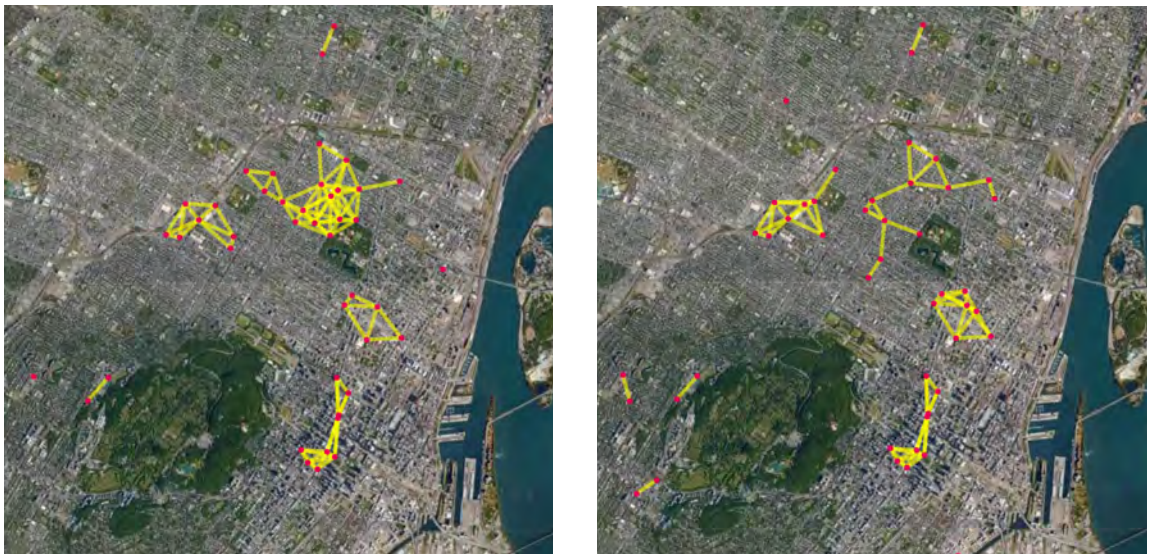
where η_i is computed as:

$$\eta_i = \frac{|\{e_{jk} \in E : n_j \in S_i, n_k \in S_i\}|}{\binom{|S_i|}{2}}. \tag{18}$$

The Watts–Strogatz clustering coefficient measures the inherent tendency of a graph to form clusters Watts and Strogatz (1998). In fact, value η_i of a node $n_i \in N$ measures how close its neighbours are to being a clique.

Strategy Pa_3 is assessed using different values of $\gamma \in \{0, 0.05, 0.10 \ldots, 1\}$ for the (α, β) combinations A, B and C obtained for Pa_2. Thus, Pa_3 is equivalent to Pa_2 when γ equals 1. Figure 6 illustrates four graphs constructed from the stations prioritized by Pa_3 in our simulation run for July 22, 2019 at 6 pm, using $\alpha = 0.5, \beta = 0.3$ and $\gamma = 0.25, 0.5, 0.75$ and 1. Nodes in N are

drawn as red dots, whereas edges in E are indicated as yellow lines. We note from the figure that small γ values yield graphs for which the selected stations can be more naturally clustered, yielding higher Watts–Strogatz clustering coefficient values.



(a) $\gamma = 0.25, \eta_G = 0.61$

(b) $\gamma = 0.5, \eta_G = 0.48$

(c) $\gamma = 0.75, \eta_G = 0.34$

(d) $\gamma = 1, \eta_G = 0.31$

Fig. 6: Stations prioritized by Pa_3 on July 22, 2019 at 6 p.m. with different values of γ.

Next, Figure 7 illustrates the impact of hyperparameter γ on the Watts-Strogatz clustering coefficient, as well as the three performance measures used in our study. In particular, for each value of γ, Figure 7a presents the average Watts-Strogatz clustering coefficient of the graphs built from the stations selected by Pa_3 during busy hours of the 2019 BIXI's season, i.e., between 7-11 am and 4-8 pm. We can notice that η_G reaches its maximum value close to $\gamma = 0.25$, and not at $\gamma = 0$. Since the stations available for prioritization might vary over time due to previous rebalancing operations, being too greedy towards the prioritization of neighbouring stations might cause an absence of clusters of unbalanced stations for subsequent simulated hours. For reference, Pa_3 yields an η_G superior to that obtained by means of the BIXI' s prioritization strategy for $\gamma \leq 0.6$.

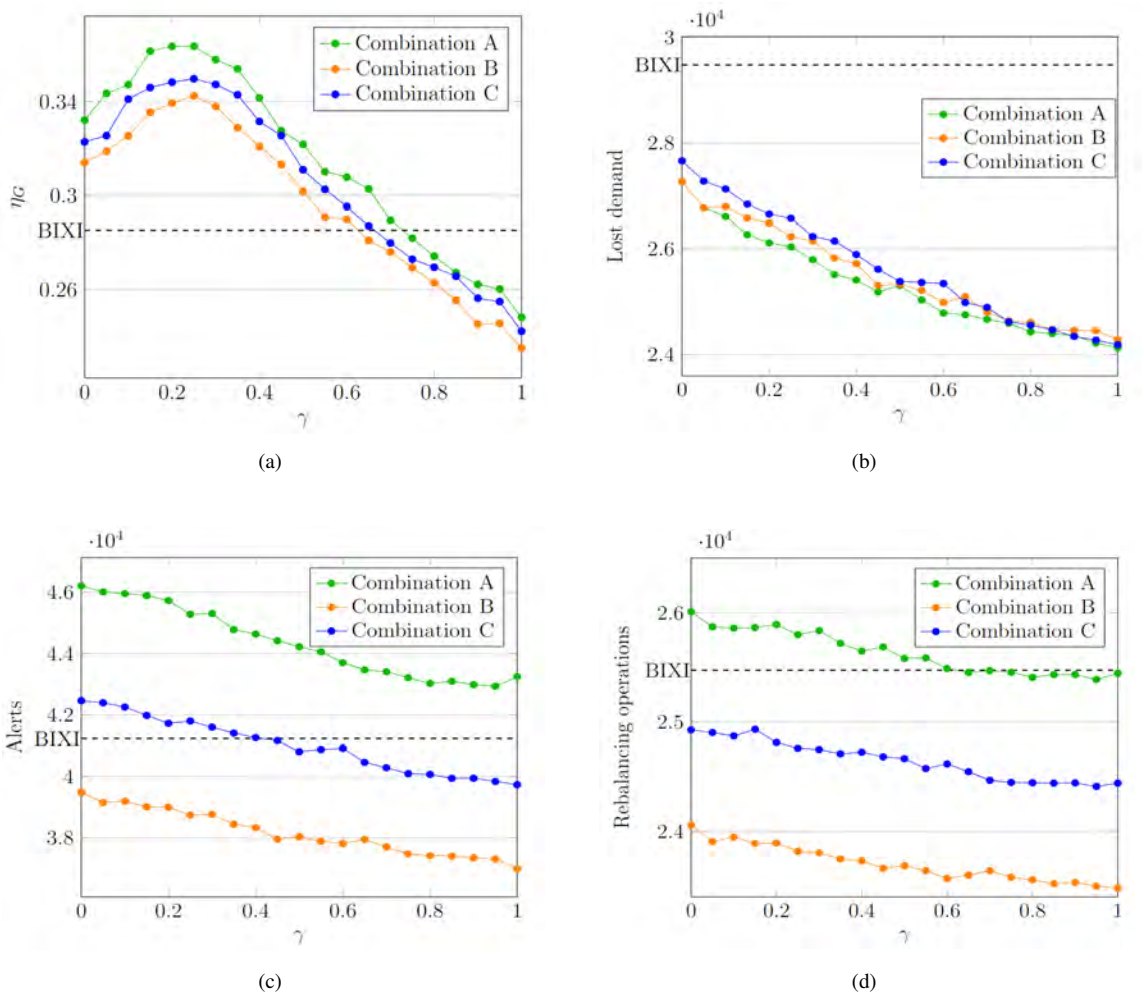


Fig. 7: Pa_3 performance measures.

Finally, for the three performance measures, Pa_3 improves its performance as γ approaches 1,

which indicates that Pa_2 is indeed the better choice when the three performance measures are the only criteria considered to select a strategy. Note, however, that the performance measures disregard the routing effort to rebalance the selected stations. Although the number of rebalancing operations observed with the use of Pa_3 is larger for smaller γ, such operations are potentially less expensive since the selected stations are closer to each other. Such tradeoff between the clustering coefficient and the cost of the system's rebalancing operations indeed merits further investigation. However, it must be conducted in conjunction with routing optimization algorithms, which are out of the scope of the present paper.

7. Concluding remarks

In this paper, we have proposed three different strategies to prioritize stations among all unbalanced stations in dock-based BSSs. This is an important issue for BSS operators, given that it is neither economically feasible nor ecologically desirable to provide a sufficiently large truck fleet in order to rebalance all unbalanced stations at the same time. From a practical perspective, these strategies are particularly attractive, given that they rely only on a predictive model and can be relatively easily implemented, therefore circumventing major technical headwinds when trying to implement rebalancing strategies based on mathematical optimization models. Once the predictive model is trained, predictions and prioritization strategies are computed in a matter of seconds, which is a practical requirement in the ongoing effort to rebalance stations throughout the day.

We have proposed a more realistic simulator that simulates the actual station inventory instead of assuming worst-case availability of bikes and empty docks. Our strategies, along with a simulation of BIXI's current rebalancing practices, have been tested on real-world data from 2019 and 2020. The results indicate that our prioritization strategies can considerably improve the number of raised alerts, lost demand and rebalancing operations. Specifically, our strategies have reduced the raised alerts up to 10%, the lost demand up to 21% and the rebalancing operations up to 12%.

Our strategies seem particularly useful during periods of high demand, when the dispatching team cannot rebalance all unbalanced stations due to capacity restrictions. Nevertheless, even for periods with low demand, our prioritization strategies successfully avoid unnecessary rebalancing operations, excluding stations that may eventually be rebalanced by the "natural" customer demand in the short-term. Among the prioritization strategies, Pa_2 outperformed Pa_1, which suggests that considering the inventory intervals (in contrast to considering only completely empty or full stations) to prioritize stations for rebalancing is important for the system performance. Algorithm Pa_3 also proved to be an effective alternative for rebalancing proximal stations, with the potential to decrease operational routing costs.

Note that the current practices at BIXI and the here proposed strategies compute the priorities for stations without distinguishing between bike drop-offs and pick-ups. This may lead to situations where the number of bikes to be picked up may be much lower or much higher than the number of bikes to be dropped off. While this issue may be partially mitigated by large truck capacities, it remains a limitation of our work. In future research, our strategies may be used along with routing optimization models, such that the entire rebalancing process is optimized in an integrated manner, taking into consideration the capacity of the dispatched trucks, as well as the loading/unloading of

bikes among the stations.

Acknowledgments

The authors are grateful to BIXI-Montreal who provided resources throughout the project. The authors also thank the Natural Sciences and Engineering Research Council of Canada (NSERC) for its financial support.

References

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. *Advances in Complex Systems* 14, 03, 415–438.

Brinkmann, J., Ulmer, M.W., Mattfeld, D.C., 2016. Inventory routing for bike sharing systems. *Transportation research procedia* 19, 316–327.

Bulhões, T., Subramanian, A., Erdoğan, G., Laporte, G., 2018. The static bike relocation problem with multiple vehicles and visits. *European Journal of Operational Research* 264, 2, 508–523.

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

Chemla, D., Meunier, F., Pradeau, T., Calvo, R.W., Yahiaoui, H., 2013b. Self-service bike sharing systems: simulation, repositioning, pricing. Technical Report hal-00824078f, Centre dâ€™Enseignement et de Recherche en MathÃ©matiques et Calcul Scientifique - CERMICS.

Chen, L., Zhang, D., Wang, L., Yang, D., Ma, X., Li, S., Wu, Z., Pan, G., Nguyen, T.M.T., Jakubowicz, J., 2016. Dynamic cluster-based over-demand prediction in bike sharing systems. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pp. 841–852.

Contardo, C., Morency, C., Rousseau, L.M., 2012. *Balancing a dynamic public bike-sharing system*, Vol. 4. Cirrelt Montreal.

Dell'Amico, M., Hadjicostantinou, E., Iori, M., Novellani, S., 2014. The bike sharing rebalancing problem: Mathematical formulations and benchmark instances. *Omega* 45, 7–19.

DeMaio, P., 2009. Bike-sharing: History, impacts, models of provision, and future. *Journal of public transportation* 12, 4, 3.

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.

Erdoğan, G., Battarra, M., Calvo, R.W., 2015. An exact algorithm for the static rebalancing problem arising in bicycle sharing systems. *European Journal of Operational Research* 245, 3, 667–679.

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.

Fricker, C., Gast, N., 2016. Incentives and redistribution in homogeneous bike-sharing systems with stations of finite capacity. *Euro journal on transportation and logistics* 5, 3, 261–291.

Gebhart, K., Noland, R.B., 2014. The impact of weather conditions on bikeshare trips in washington, dc. *Transportation* 41, 6, 1205–1225.

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

Hampshire, R.C., Marla, L., 2012. An analysis of bike sharing usage: Explaining trip generation and attraction from

observed demand. In *91st Annual meeting of the transportation research board, Washington, DC*, pp. 12–2099.

Hulot, P., 2018. Towards station-level demand prediction for effective rebalancing in bike-sharing systems. Master's thesis, Ã‰cole Polytechnique de MontrÃ©al.

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.

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 *Twenty-seventh international conference on automated planning and scheduling*.

Lu, Y., Benlic, U., Wu, Q., 2020. An effective memetic algorithm for the generalized bike-sharing rebalancing problem. *Engineering Applications of Artificial Intelligence* 95, 103890.

Mellou, K., Jaillet, P., 2019. Dynamic resource redistribution and demand estimation: An application to bike sharing systems. *SSRN Electron* pp. 1–58.

Nunes, P., Moura, A., Santos, J., 2022. Solving the multi-objective bike routing problem by meta-heuristic algorithms. *International Transactions in Operational Research*

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

Papazek, P., Kloimüllner, C., Hu, B., Raidl, G.R., 2014. Balancing bicycle sharing systems: an analysis of path relinking and recombination within a grasp hybrid. In *International Conference on Parallel Problem Solving from Nature*, Springer, pp. 792–801.

Papazek, P., Raidl, G.R., Rainer-Harbach, M., Hu, B., 2013. A pilot/vnd/grasp hybrid for the static balancing of public bicycle sharing systems. In *International Conference on Computer Aided Systems Theory*, Springer, pp. 372–379.

Pase, F., Chiariotti, F., Zanella, A., Zorzi, M., 2020. Bike sharing and urban mobility in a post-pandemic world. *IEEE Access* 8, 187291–187306.

Pucher, J., Buehler, R., Bassett, D.R., Dannenberg, A.L., 2010. Walking and cycling to health: a comparative analysis of city, state, and international data. *American journal of public health* 100, 10, 1986–1992.

Ren, Y., Meng, L., Zhao, F., Zhang, C., Guo, H., Tian, Y., Tong, W., Sutherland, J.W., 2020. An improved general variable neighborhood search for a static bike-sharing rebalancing problem considering the depot inventory. *Expert Systems with Applications* 160, 113752.

Rosenthal, R.E., 1985. Concepts, theory, and techniques principles of multiobjective optimization. *Decision Sciences* 16, 2, 133–152.

Schuijbroek, J., Hampshire, R.C., Van Hoeve, W.J., 2017. Inventory rebalancing and vehicle routing in bike sharing systems. *European Journal of Operational Research* 257, 3, 992–1004.

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. *Operations Research* 61, 6, 1346–1359.

Tan, B., Karabati, S., 2004. Can the desired service level be achieved when the demand and lost sales are unobserved? *IIE Transactions* 36, 4, 345–358.

Vallez, C.M., Castro, M., Contreras, D., 2021. Challenges and opportunities in dock-based bike-sharing rebalancing: a systematic review. *Sustainability* 13, 4, 1829.

Vergeylen, N., Sörensen, K., Vansteenwegen, P., 2020. Large neighborhood search for the bike request scheduling problem. *International Transactions in Operational Research* 27, 6, 2695–2714.

Wang, M., Zhou, X., 2017. Bike-sharing systems and congestion: Evidence from us cities. *Journal of transport geography* 65, 147–154.

Wang, S., He, T., Zhang, D., Shu, Y., Liu, Y., Gu, Y., Liu, C., Lee, H., Son, S.H., 2018. Bravo: Improving the rebalancing operation in bike sharing with rebalancing range prediction. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 1, 1–22.

Watts, D.J., Strogatz, S.H., 1998. Collective dynamics of â€˜small-worldâ€™networks. *Nature* 393, 6684, 440–442.

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