EURO Journal on Transportation and Logistics 13 (2024) 100140

Contents lists available at ScienceDirect

EURO Journal on Transportation and Logistics

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

A two-stage stochastic programming model for bike-sharing systems with rebalancing

Rossana Cavagnini [a,*], Francesca Maggioni [b], Luca Bertazzi [c], Mike Hewitt [d]

[a] *Deutsche Post Chair – Optimization of Distribution Networks, RWTH Aachen University, Germany*
[b] *Department of Management, Information and Production Engineering, University of Bergamo, Italy*
[c] *Department of Economics and Management, University of Brescia, Italy*
[d] *Department of Information Systems and Operations Management, Quinlan School of Business, Loyola University Chicago, USA*

A R T I C L E I N F O

Keywords:
Bike sharing
Target inventory levels
Rebalancing
Stochastic programming
Analysis of the stochastic solution

A B S T R A C T

We study the problem of determining the target inventory level of stations in a bike-sharing system, when bikes can be rebalanced later during the day. We propose a two-stage stochastic programming formulation, where the target inventory decisions are made at the first stage, while the recourse decisions, related to rebalancing, are made at the second stage. In the literature, the problem of determining the target inventory levels is solved without taking into account the rebalancing problem, or these two problems are solved sequentially. We prove that more efficient bike-sharing systems can be obtained by integrating these two problems. Moreover, we show that our methodology provides better results than the deterministic formulation, and consider an effective matheuristic, based on the solution of the deterministic problem, to solve the stochastic program. Finally, we compare the solutions obtained by our approach with the actual allocation of bikes in the real bike-sharing system of the city of San Francisco. The results show the effectiveness of our approach also in a realistic setting.

1. Introduction

Bike-sharing systems are becoming more popular throughout the world, doubling their number from 550 in 2012, to more than 3000 in 2021 (PBSC, 2021). The popularity of these systems can be attributed to an increasing interest in reducing pollution and traffic, as well as promoting healthy lifestyles, worldwide. Bike-sharing systems provide a fleet of bikes used by different individuals throughout the day, usually via a rental agreement. These systems typically consist of a depot (or a set of depots), wherein bikes are stored at the beginning of the day, and multiple stations located throughout the city. An individual can withdraw a bike from one of these stations, to make a (usually short) journey, and then return that bike to a possibly different station. These stations have a fixed number of slots, even if this number can sometimes be temporarily increased. A technology, to communicate information regarding their status (e.g., how many bikes are currently there) to a central manager/planner, is typically available at these stations. Bike-sharing systems are financed by public and/or private entities and managed by service providers, who are involved in strategic, tactical, and operational decision-making. Strategic decisions involve determining the number, location, and capacity of stations, whereas tactical decisions concern fleet sizing and allocation decisions. Daily operational decisions include determining how to periodically re-distribute bikes to stations.

This paper studies a bike-sharing system composed of one depot (with an initial availability of bikes) and multiple capacitated stations. The capacity of the stations can be enlarged by individuals who can accept returning bikes even when the station is at capacity. This option is called "valet service" and is implemented by multiple bike-sharing systems throughout the world, such as the one of San Francisco (S.F. Gov, 2024) and Chicago (Divvy Bikes, 2024). These bike-sharing systems allocate staff members to some stations, who can remove bikes from docks and park them on the sidewalk or between docks. This guarantees the availability of additional free slots, where users can return their bikes even when the number of bikes exceeds the capacity. As highlighted by de Chardon et al. (2016), the valet service improves the system reliability by guaranteeing the possibility of returning bikes. In fact, one of the reasons for users not to renew their subscriptions is the anxiety of not finding a free slot for returning bikes.

The service provider has to decide the target inventory level of each station, considering that bikes can be rebalanced among stations

* Corresponding author.
E-mail addresses: cavagnini@dpo.rwth-aachen.de (R. Cavagnini), francesca.maggioni@unibg.it (F. Maggioni), luca.bertazzi@unibs.it (L. Bertazzi), mhewitt3@luc.edu (M. Hewitt).

https://doi.org/10.1016/j.ejtl.2024.100140
Received 8 September 2023; Received in revised form 29 May 2024; Accepted 15 July 2024
Available online 22 July 2024

later during the day. Rebalancing is performed through a capacitated vehicle that travels a given route. The service provider tries to limit the cases in which, after performing rebalancing, an individual arrives at a station for renting a bike, but none is available. In inventory management, these situations represent stock-out events and we refer them to as "*starvation*" events. On the other hand, the service provider tries to limit the cases in which, after performing rebalancing, an individual seeks to return a bike to a station but the station is already full. In inventory management, these situations are similar to what happens in order-up-to-level and maximum-level systems. Once the maximum inventory level (i.e., the capacity of the station) is reached, no additional units can be received until at least one unit is delivered. We refer these situations to as "*congestion*" events. Both starvation and congestion events negatively impact the user's experience, as they both (potentially) require the user to travel to another station. At the same time, avoiding starvation events and avoiding congestion events are somewhat competing objectives, as the more bikes allocated to a station, the lower the likelihood of a starvation event, but the greater the likelihood of a congestion event. Apart from congestion and starvation, the service provider may also seek to limit the size of the bike fleet in use (to prevent it from damage and deterioration), as well as the number of bikes that are redistributed throughout the day (to limit rebalancing costs).

In the literature, there are papers proposing approaches to determine the target inventory level of stations without considering rebalancing operations. These approaches may result in overestimations or underestimations of the target inventory levels and, consequently, in a higher number of congestion and starvation events. To fill this gap, we propose an approach in which rebalancing is considered when these levels are computed. On the other hand, there are papers that consider some measures of bike redistribution, but either they study the problem from the deterministic point of view or lack inclusion of realistic bike-sharing system features. To fill these gaps, we propose an optimization model that not only accounts for the uncertainty in the bike demand, but also incorporates some of these realistic features of bike-sharing systems, such as the presence of the valet service and the route-based rebalancing operations.

In particular, we formulate a new two-stage Stochastic Program (SP) that models the problem described above. The purpose of this model is to determine an effective target level for all stations (first stage solution), approximating the cost of rebalancing bikes with a static second-stage recourse. We assume that rebalancing is performed by visiting stations according to a sequence determined by a pre-computed route. As such, the estimation of the rebalancing cost depends only on the decision of the number of bikes to move from one station to one of the next (second stage decision). Notice that, as highlighted in Erera et al. (2009), a fixed route is also more desirable operationally. Our SP includes multiple performance dimensions (i.e., congestion, starvation, fleet size, and rebalancing) and it can be solved multiple times in a day. To accurately describe the uncertain parameters involved in this problem, we propose a method to generate realistic scenarios and establish the minimal dimension of the corresponding scenario tree. Moreover, we propose a new approach to compute the demand which allows to capture its evolution over time. Several indicators of anticipating rebalancing, of the value of the SP solution and of the quality of the deterministic solutions are considered. In particular, through an analysis of the deterministic solution, we consider a matheuristic, based on the deterministic solution, to solve our SP. Moreover, we design a simulation framework that is run to compare the quality of different bike allocation plans along multiple performance dimensions. Finally, with a sensitivity analysis, we assess the behavior of the first and second-stage cost for increasing values of congestion and starvation penalties.

Our computational experiments are based on a set of instances of the real bike-sharing system of San Francisco. We first establish computationally that our modeling approach leads to better solutions than the one in which the demand is considered to be static. The results also suggest that better allocation plans can be obtained by considering rebalancing when the target inventory levels at stations are computed. Moreover, we show that explicitly modeling uncertainty in demand leads to improvements along multiple performance dimensions over using a deterministic model. Nevertheless, the results also indicate that using information from the deterministic solution within a matheuristic framework allows us to find good-quality solutions for the SP in short runtimes. Finally, the results reveal that the allocation plan provided by our SP outperforms the one of the real San Francisco bike-sharing system, on all dimensions.

The paper is organized as follows. In Section 2, we discuss the relevant literature, and contrast both the problems studied and methodologies with the research proposed in this paper. In Section 3, we describe the problem, while the SP is formulated in Section 4. In Section 5, we describe all methods. In Section 6, we show the computational results. Finally, Section 7 provides conclusions and suggestions for future works.

2. Literature review

In this section, we present a review of the papers studying station-based bike-sharing systems. While reviewing the literature, we observed that works that only focus on determining the optimal allocation of bikes to stations (without rebalancing considerations) use the term "initial inventory", while a large part of the works that considers only rebalancing or the optimal allocation and rebalancing use the term "target inventory". In both cases, these terms represent the number of bikes that should ideally be at stations. For consistency, in the following, we refer to these quantities as "target inventory".

The reviewed papers are summarized in Table 1 and can be classified into: (i) papers focusing exclusively on the problem of determining the target inventory level at stations (Section 2.1), (ii) papers studying exclusively the problem of rebalancing (Section 2.2), and (iii) papers considering both problems jointly (Section 2.3).

For a review of additional bike-sharing problems (including, for example, problems at the strategic decision level, problems dealing with free-floating bike-sharing systems, and problems considering bike-sharing incentives and parking reservation schemes), the reader can refer to the work of Shui and Szeto (2020). In Section 2.4, we conclude with a summary of our contributions with respect to the extant literature.

2.1. Target inventory level problems

Raviv and Kolka (2013) focus on the problem of determining the optimal target inventory level at a single bike-sharing station. They minimize a user dissatisfaction measure, expressed as a function of the bike and locker shortage events. The proposed model omits interdependencies among stations, both with reference to the withdrawn and returned bike processes, and the penalties charged which are the same across all stations. To overcome these limitations, Datner et al. (2017) study the problem of setting the optimal target inventory levels in a bike-sharing system with interactions between stations. These interactions are captured by a user-behavior model which describes the decisions a user may make in response to a shortage of bikes or lockers, that are: waiting at a station, roaming to a nearby station, or abandoning the system. The objective is to minimize the excess time spent by a user in the system because of a shortage of bikes or lockers. While this work is the first to consider interactions between stations in setting the initial inventory levels and, to the best of our knowledge, it is the only one that considers a stochastic demand process, it presents three differences with respect to our work: (i) it does not consider the capacity of the stations and the presence of a possibly limited bike fleet size (i.e., some inventory levels could not be implementable), (ii) it does not consider the possibility that rebalancing may take place,

Table 1
Bike-sharing literature classification based on the problem features and the solution methodology.

Article	Problem features			Solution method
	Target inventory	Rebalancing	Stochastic (S), Deterministic (D)	
Raviv and Kolka (2013)	✓		S	exact
Datner et al. (2017)	✓		S	heuristic
Benchimol et al. (2011)		static	D	exact
Chemla et al. (2013)		static	D	heuristic
Raviv et al. (2013)		static	D	exact
Erdoğan et al. (2014)		static	D	exact
Ho and Szeto (2014)		static	D	heuristic
Dell'Amico et al. (2014)		static	D	exact
Forma et al. (2015)		static	D	heuristic
Erdoğan et al. (2015)		static	D	exact
Alvarez-Valdes et al. (2016)		static	D	heuristic
Dell'Amico et al. (2016)		static	D	heuristic
Cruz et al. (2017)		static	D	heuristic
Nair and Miller-Hooks (2011)		static	S	exact and heuristic
Dell'Amico et al. (2018)		static	S	exact and heuristic
Bulhões et al. (2018)		static	D	exact and heuristic
Bruck et al. (2019)		static	D	exact
Lv et al. (2020)		static	D	heuristic
Contardo et al. (2012)		dynamic	D	heuristic
Brinkmann et al. (2015)		dynamic	S	heuristic
Brinkmann et al. (2016)		dynamic	D	heuristic
Ghosh et al. (2017)		dynamic	D	heuristic
Brinkmann et al. (2019)		dynamic	S	heuristic
Legros (2019)		dynamic	D	heuristic
Li et al. (2024)		dynamic	S	heuristic
Vogel et al. (2014)	✓	dynamic	D	heuristic
Regue and Recker (2014)	✓	dynamic	D	heuristic
Neumann-Saavedra et al. (2015)	✓	dynamic	D	exact
Lu (2016)	✓	static	S	exact
Vogel (2016)	✓	dynamic	D	heuristic
Schuijbroek et al. (2017)	✓	static	S	heuristic
Maggioni et al. (2019)	✓	static	S	exact
Ren et al. (2020)	✓	static	D	heuristic
Fu et al. (2022)	✓	dynamic	S	heuristic
Our paper	✓	static	S	exact and heuristic

and (iii) it does not consider that the service providers might also be interested in minimizing the operational costs related to the allocation of bikes at stations. Differently from these two papers, in our work, the interdependence among stations is captured by making penalties for congestion and starvation events dependent on the distance to the next-closest station or on the average distance to all other stations.

2.2. Rebalancing problems

Concerning rebalancing, we can divide the papers into two different groups depending on how often the repositioning of bikes is performed, i.e., static rebalancing and dynamic rebalancing.

2.2.1. Static rebalancing

With static rebalancing, we refer to rebalancing operations performed at night, when the system is closed or idle, to prepare the system for the next day. According to Berbeglia et al. (2007), such a problem is similar to the one-to-one pick-up and delivery problem (PDP) with transshipment, since each rebalancing flow has exactly one pick-up station and one delivery station, and these stations are determined by the order in which they are visited.

However, this variant of the PDP does not consider the impact on the customer service level (e.g., dissatisfaction due to congestion and starvation events). The same limitation can be found in the following papers which also use predetermined target levels (or ranges) for the number of bikes that should be at each station after rebalancing. Benchimol et al. (2011) formulate a modified version of the capacitated traveling salesman problem to study the problem of finding a minimal route that balances all stations, and they use an approximation algorithm to address the problem. Chemla et al. (2013) propose some relaxations to derive good lower bounds and a tabu search heuristic to get upper bounds for the static rebalancing problem. Erdoğan et al. (2014, 2015) propose exact methods to solve the problems. Similarly to our work, the first assumes that rebalancing is done via a single vehicle following a fixed route. Cruz et al. (2017) study a rebalancing problem where only a single vehicle is available, but multiple visits to the same station are allowed and they propose an iterated local search heuristic. Dell'Amico et al. (2014) formulate models relying on different pick-up and delivery problems and solve these formulations via a branch-and-cut algorithm, while (Dell'Amico et al., 2016) extend their work by proposing a destroy and repair algorithm. Recently, Bruck et al. (2019) propose multiple model versions and exact algorithms to perform rebalancing while minimizing the rebalancing costs measured in terms of distance. They also impose that stations cannot be used as temporary depots to provisionally collect and store bikes. In all three of the latter papers, the objective is to minimize rebalancing costs using a fleet of capacitated vehicles, instead of a single one. Besides the presence of multiple vehicles, Alvarez-Valdes et al. (2016) and Bulhões et al. (2018) also allow multiple visits to stations. Lv et al. (2020) introduces the presence of multiple depots for rebalancing.

A number of papers also considers the customer satisfaction dimension. Raviv et al. (2013) propose a deterministic multi-objective mathematical program in which they minimize an objective that consists of penalties for stockouts, penalties for stations being at capacity when users wish to return bikes, and the operational costs incurred when rebalancing. They present an arc-indexed and a time-indexed formulation, with the first limiting each vehicle to visit each station at most once. Their first formulation is also studied in Forma et al. (2015), but solved using a matheuristic based on the clustering-first routing-second paradigm, while Ho and Szeto (2014) propose a modification of the problem studied in Forma et al. (2015), by not considering the rebalancing costs.

Among stochastic static rebalancing problems, we mention the work of Nair and Miller-Hooks (2011) and that of Dell'Amico et al. (2018), who propose multiple stochastic programming formulations to deal with uncertain demand by deciding vehicle routes and rebalanced quantities. Apart from minimizing the rebalancing costs, they also consider penalties for each unit of deviation with respect to the stations target inventory levels.

2.2.2. Dynamic rebalancing

Even if much of the literature focuses on static rebalancing, some works have focused on its dynamic version. With dynamic rebalancing, we refer to rebalancing operations executed during the day when the system is in use and, hence, unexpected bike withdrawals or returns may occur while rebalancing is performed.

Examples of works studying dynamic rebalancing and focusing on deterministic demand are those by Contardo et al. (2012), Ghosh et al. (2017), and Brinkmann et al. (2016). Among the works dealing with stochastic dynamic rebalancing, Brinkmann et al. (2015) minimizes the expected number of violations of due dates, i.e., the latest time a station has to be served by a vehicle to satisfy a request. Brinkmann et al. (2019) propose a stochastic-dynamic inventory routing problem for rebalancing bikes with one vehicle, and Brinkmann et al. (2020) extends this work to the multi-vehicle case. Legros (2019) adopts a Markov decision process approach based on a decomposition at a station level to decide which station should be prioritized and the amount of bikes to move between stations with the objective of minimizing user dissatisfaction. Recently, Li et al. (2024) formulate a multi-period two-stage stochastic model for the dynamic rebalancing problem with stochastic demand. The authors propose a rolling horizon framework within which a hybrid metaheuristic (based on genetic algorithms and variable neighborhood search) is executed.

2.3. Target inventory level problems with rebalancing

Finally, the following papers consider the problem of determining the number of bikes at stations and of rebalancing jointly.

Vogel et al. (2014), Neumann-Saavedra et al. (2015), and Vogel (2016) rely on a service network design formulation considering both the problem of determining the station optimal target inventory levels, and of dynamically rebalancing. Regue and Recker (2014) present a sequential framework for making target inventory level and rebalancing decisions. The first step in this framework is to forecast demand at each station, from which an initial inventory level for that station is determined. The second step is to determine a dynamic rebalancing plan based upon those target inventory levels, which dictate how many bikes should be transported from one station to another. Then, the third step is to determine vehicle routes to execute that rebalancing plan. Lu (2016) studies a problem that focuses on both the target inventory level and rebalancing of bikes, wherein there is uncertainty in how bike usage will deviate from what is expected. Their objective function measures bike supply cost, inventory and redistribution costs, and penalties associated with stock-outs. Differently from our approach, they propose a robust model of this problem that seeks to minimize their objective under a maximum demand scenario generated from two different uncertainty sets. A robust optimization approach is also adopted by Fu et al. (2022), who maximizes the revenue minus the total rebalancing costs. Schuijbroek et al. (2017) propose a sequential approach for solving the problem of determining the service level requirements (i.e., target inventory intervals) at each station and the rebalancing vehicle routes. They model the stochastic demand by considering the inventory at each station independently as a non-stationary queuing system with finite capacity. Uncertain demands are also considered in Maggioni et al. (2019), who formulate a two-stage and a multi-stage stochastic program to determine the target initial inventory level at stations to minimize the sum of bike procurement costs, expected bike and locker stockout costs, and transshipment costs for performing static rebalancing of bikes. In their work, rebalancing is performed under the assumption that bikes can be transferred directly between two locations, i.e., without following a route. This assumption is reasonable for bike-sharing systems where the duration of rebalancing is not critical, as direct bike transfers usually result in longer times and higher costs compared to those that follow a route. Finally, Ren et al. (2020) study the static rebalancing problem with multiple vehicles and stations which can only be visited once. Apart from minimizing the variable and fixed rebalancing costs, the authors also focus on the fleet size, i.e., on the number of bikes leaving the depot which are initially loaded by each vehicle.

2.4. Contribution of this work

In light of the literature reviewed, we believe our work makes the following contributions. Compared to the literature that focuses only on determining target inventory levels, or considers the target inventory level problem and rebalancing sequentially, our approach provides better target inventory levels because of the simultaneous consideration of both inventory levels and rebalancing decisions. This approach leads to fewer underestimations and overestimations of target inventory levels, resulting in a more efficient system with fewer congestion and starvation events.

Moreover, unlike papers that exclusively focus on rebalancing operations, our goal is to determine the ideal number of bikes that should be at stations and that should be used as a target for rebalancing operations. Therefore, these targets are decision variables in our problem, rather than given parameters.

With respect to the papers that study how to determine target inventory levels by considering rebalancing operations in a deterministic setting, we propose an approach that explicitly considers the uncertainty in the demand. With our approach, we obtain better target inventory levels that improve the supply and demand match for the system.

Finally, compared to the papers that recognize uncertainty in the demand, our approach includes realistic features of bike-sharing systems, such as the presence of the valet service, or route-based rebalancing operations. Motivated by the increased complexity arising from considering a stochastic model, we further consider a matheuristic approach to obtain high-quality solutions for the stochastic program with reduced runtimes.

3. Problem description

We study a bike-sharing system managed by a service provider wherein the decision-making is centralized. The service provider has to determine the target inventory level of bikes to allocate at each station under uncertain demand, considering that bikes are rebalanced at a later point in time to restore that target inventory level.

We consider a single depot, multiple stations with given capacities, and a number of bikes that can already be at stations. We assume that there is a limited number of bikes corresponding to the depot capacity. We presume rebalancing is executed by a capacitated vehicle that travels along a known and fixed route (determined *a priori* by solving a Traveling Salesman Problem) that begins at the depot, visits each station, and ends at the depot.

Because our goal is to make better allocation decisions by considering an approximation of the rebalancing costs, it is reasonable to consider such a fixed route. In fact, we do not aim to obtain the exact rebalancing plan that will be executed, but just an estimate of the rebalancing costs. Because of this, we also do not model the possibility to move bikes from the depot to a station during the rebalancing operation. This choice allows us to measure the ideal fleet size as the sum over all stations of their target inventory levels. However, our rebalancing cost approximation considers that bikes can still be redistributed from the first station to the next stations visited in the tour.

The capacity of each station can be temporarily expanded using the so-called "valet service", which consists of individuals who can keep station docks free when the station would be at capacity. We assume that, if the rebalancing vehicle fails to bring the number of bikes at a station below its capacity, it dispatches a staff member to that station. Hence, the valet service is available everywhere whenever it is needed. The managing effort of this staff member is dependent on the number of bikes in excess at that station.

Because the service provider wants to maximize the customer service level, the occurrence of both congestion (a user wishes to return a bike to a station but it is full) and starvation (a user wishes to rent a bike from a station but it is empty) must be avoided. At the same time, the provider wants to minimize operational costs. For this, the number of bikes allocated and rebalanced (to prevent bike damage) is minimized. Moreover, the number of bikes exceeding the station capacity must be small to decrease the likelihood of resorting to the valet service. The objective of our problem includes all these aspects.

4. A two-stage stochastic programming (SP) formulation

In this section, we propose a two-stage SP of the problem presented in Section 3. We refer to Birge and Louveaux (2011) and King and Wallace (2012) for comprehensive books on Stochastic programming. Table 7 in Appendix A summarizes the used notation.

We denote the set of bike-stations by $\mathcal{I} = \{1, \dots, I\}$, where I is the depot, and the set of possible realizations of uncertainty, i.e. scenarios, by $\mathcal{S} = \{1, \dots, S\}$.

The depot capacity and the total availability of bikes correspond to $\overline{I}_{I0}$, while $\overline{I}_{i0}$ represent the initial availability of bikes at station $i \in \mathcal{I} \setminus \{I\}$. The capacity of each station is denoted by Q_i, and the capacity of the vehicle used for rebalancing is represented by C.

The demand at each station is measured as the difference between the uncertain number of withdrawn and returned bikes at that station, during the period between when bikes are ideally allocated and redistributed. To mitigate the limitation of the static demand assumption, we encourage the model to allocate no fewer bikes than the maximum number of consecutive withdrawn bikes from a station before a return event, and to maintain no fewer free docks than the maximum number of consecutive bikes returned to a station before a withdrawal event. This approach allows us to prevent congestion and starvation especially if there are peaks of consecutive withdrawn and returned bikes. As an example, suppose that there is a maximum of two bikes consecutively withdrawn from a station before a return occurs, in a given time interval. By only considering the net demand, the model will suggest a target inventory level of one bike $(2 - 1)$. This results in a starvation event when a user wants to withdraw the second bike. However, with our approach, the model is encouraged to set a target inventory level of two (maximum number of withdrawn bikes before a return occurs). This allows us to prevent a starvation event.

We define d_i^s as the stochastic demand of bikes at station $i \in \mathcal{I} \setminus \{I\}$ in scenario $s \in \mathcal{S}$. Moreover, g_i^s and h_i^s represent the stochastic maximum number of consecutive bikes withdrawn from station $i \in \mathcal{I} \setminus \{I\}$ before a return occurs in scenario $s \in \mathcal{S}$, and the stochastic maximum number of consecutive bikes returned to station $i \in \mathcal{I} \setminus \{I\}$ before a withdrawal occurs in scenario $s \in \mathcal{S}$, respectively. The probability of scenario $s \in \mathcal{S}$ is denoted by pr^s.

Before the stochastic parameters (i.e., d_i^s, g_i^s, and h_i^s) become known, the service provider needs to determine the value of the non–negative integer variables x_i, i.e., the target inventory level of station $i \in \mathcal{I} \setminus \{I\}$. These are the first-stage variables of our SP. A unit penalty cost f_i is charged, to take into account the cost of the potential damage of each used bike.

After allocating the bikes, at a later point during the day, the stochastic demands d_i^s, and the stochastic maximum number of consecutive withdrawn and returned bikes g_i^s and h_i^s are observed at each station i. Then, the service provider determines a rebalancing plan based on the number of bikes available at each station. We indicate the number of bikes to rebalance from station i to station $i+1$ in scenario s by the integer variables $y_{i,i+1}^s$. These are the second-stage variables. Each rebalanced bike is penalized by $t_{i,i+1}$. After rebalancing, the surplus or shortage at each station can be immediately computed. We denote the balance of bikes at station $i \in \mathcal{I} \setminus \{I\}$ in scenario $s \in \mathcal{S}$ by $I_i^s \in \mathbb{Z}$, while the units of surplus and stock-out at station $i \in \mathcal{I} \setminus \{I\}$ in scenario $s \in \mathcal{S}$ are represented by I_i^{s+} and I_i^{s-}, respectively. Moreover, B_i^s indicates the extra inventory balance, and E_i^s the excess inventory balance at station $i \in \mathcal{I} \setminus \{I\}$ in scenario $s \in \mathcal{S}$.

We recall that in our problem, the service provider seeks to avoid the occurrence of both congestion (a user wishes to return a bike to a station but it is full) and starvation (a user wishes to rent a bike from a station but it is empty).

Congestion is measured by the "extra inventory" term B_i^{s+}, which represents the number of bikes at station $i \in \mathcal{I} \setminus \{I\}$ in scenario s that is above and beyond the initial inventory plus the allocated bike number. Recalling the possible need for the valet service, with the "excess inventory" term E_i^{s+}, $i \in \mathcal{I} \setminus \{I\}$, we measure the number of bikes in excess of station capacity in scenario s. We refer to the weight associated with "excess inventory" as the "excess penalty" c_i, $\forall i \in \mathcal{I} \setminus \{I\}$ and the weight associated with "extra inventory" as the "extra penalty" $\frac{c_i}{Q_i}$, $\forall i \in \mathcal{I} \setminus \{I\}$. Because the extra inventory represents an increased risk of incurring congestion, it is penalized in relative terms with respect to the capacity of each station. In fact, given the same extra inventory level at two stations characterized by two different capacities, the risk of congestion is higher for the station with lower capacity and our penalty c_i/Q_i represents the magnitude of this risk.

Starvation is measured by I_i^{s-}, $i \in \mathcal{I} \setminus \{I\}$, representing the realized shortage of bikes at station i in scenario s. We refer to the weight associated with starvation as the "stock-out penalty" p_i, $i \in \mathcal{I} \setminus \{I\}$. Note that we do not penalize the risk of incurring starvation, i.e., we do not penalize the number of bikes that is below the sum of the initial inventory and the allocated bikes (unless we encounter stock-out). In fact, because congestion events are more frustrating than starvation events for users, operators prioritize congestion avoidance over starvation (de Chardon et al., 2016).

We finally recall that the stochastic parameters g_i^s and h_i^s are introduced to encourage the model to determine a target inventory level higher than g_i^s and to maintain no fewer free docks than h_i^s. To guarantee that a feasible solution can be found if the sum of the maximum number of consecutive withdrawn and returned bikes is greater than the station capacity (i.e., $g_i^s + h_i^s > Q_i$), we introduce the variables a_i^s to represent the slack between the number of allocated bikes x_i and g_i^s at station $i \in \mathcal{I} \setminus \{I\}$ in scenario $s \in \mathcal{S}$. Similarly, the variables b_i^s denote the units of slack between the number of free racks $Q_i - x_i$ and h_i^s at station $i \in \mathcal{I} \setminus \{I\}$ in scenario $s \in \mathcal{S}$.

We illustrate the sequence of decisions and events with an example in Fig. 1. The leftmost part of Fig. 1 represents the first-stage decision, in which bikes are allocated to stations according to the target inventory level (see blue arrows) determined before knowing the realized demand, and the realized maximum number of consecutive withdrawn and returned bikes before a return or withdrawal event occurs. In the central part of the figure, each orange arrow represents one scenario describing the bike demand, and the maximum number of consecutive withdrawn and returned bikes at stations. Finally, in the rightmost part of Fig. 1, each small figure shows a recourse decision to make for each scenario that may have realized. Each of these figures represents the number of bikes rebalanced by a vehicle following a fixed route described by green arrows. Hence, the rightmost part of Fig. 1 represents the rebalancing cost approximation, that depends on the first-stage decision, and on the stochastic demand and maximum number of withdrawn and returned bikes.

As such, this model can be solved repeatedly throughout the day. For example, at 6:00 a.m., the service provider wants to determine the target inventory levels of stations. For this, they solve the SP which

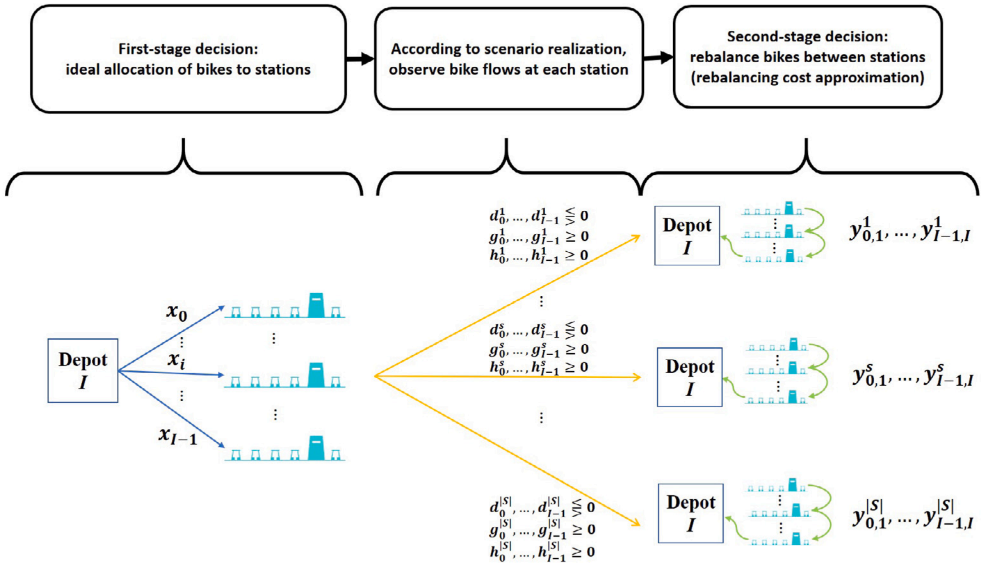


Fig. 1. Illustration of the two-stage decision-making process.

returns the number of bikes which should ideally be at each station by considering an approximation of the costs occurring for rebalancing at 12:00 p.m. Note that our approximation is based on a rebalancing plan that will not necessarily be the one that will actually be implemented at 12:00 p.m. At 6:00 a.m., the service provider makes sure that each station has the number of bikes prescribed by the solution of our SP. Then, at 12:00 p.m., the service provider solves again the SP to get the new target inventory level of stations at noon. This time, this decision is based on an approximation of the costs that will be charged for rebalancing at 6:00 p.m.

The problem can be formulated as the following integer non-linear SP, which we linearize in Appendix B: Problem $\mathcal{B}$

$$\min \sum_{i\in\mathcal{I}} f_i x_i + \sum_{s\in\mathcal{S}} pr^s [\sum_{i\in\mathcal{I}\setminus\{I\}} (t_{i,i+1} y^s_{i,i+1} + \frac{c_i}{Q_i} B_i^{s+} + c_i E_i^{s+} + p_i(-I_i^{s-}) + p_i a_i^s + c_i b_i^s)] \tag{1}$$

s.t:

$$x_i \geq g_i^s - a_i^s \qquad i \in \mathcal{I}\setminus\{I\},\ s\in\mathcal{S}, \tag{2}$$

$$Q_i - x_i \geq h_i^s - b_i^s \qquad i \in \mathcal{I}\setminus\{I\},\ s\in\mathcal{S}, \tag{3}$$

$$\overline{I}_{i0} + x_i \leq Q_i \qquad i \in \mathcal{I}\setminus\{I\}, \tag{4}$$

$$\sum_{i\in\mathcal{I}\setminus\{I\}} x_i \leq \overline{I}_{I0}, \tag{5}$$

$$y^s_{i,i+1} \leq C \qquad i \in \mathcal{I}\setminus\{I\},\ s\in\mathcal{S}, \tag{6}$$

$$I_I^s = \overline{I}_{I0} - \sum_{i\in\mathcal{I}\setminus\{I\}} x_i + y^s_{I-1,I} \qquad s\in\mathcal{S}, \tag{7}$$

$$I_I^s \leq \overline{I}_{I0} \qquad s\in\mathcal{S}, \tag{8}$$

$$I_1^s = \overline{I}_{i0} + x_1 - d_1^s - y^s_{1,2} \qquad s\in\mathcal{S}, \tag{9}$$

$$I_i^s = \overline{I}_{i0} + x_i - d_i^s + y^s_{i-1,i} - y^s_{i,i+1} \qquad i \in \mathcal{I}\setminus\{1, I\},\ s\in\mathcal{S}, \tag{10}$$

$$I_i^{s+} = \max\{0, I_i^s\} \qquad i \in \mathcal{I}\setminus\{I\},\ s\in\mathcal{S}, \tag{11}$$

$$I_i^{s-} = \min\{0, I_i^s\} \qquad i \in \mathcal{I}\setminus\{I\},\ s\in\mathcal{S}, \tag{12}$$

$$E_i^s = I_i^{s+} - Q_i \qquad i \in \mathcal{I}\setminus\{I\},\ s\in\mathcal{S}, \tag{13}$$

$$E_i^{s+} = \max\{0, E_i^s\} \qquad i \in \mathcal{I}\setminus\{I\},\ s\in\mathcal{S}, \tag{14}$$

$$B_i^s = I_i^{s+} - x_i - \overline{I}_{i0} - E_i^{s+} \qquad i \in \mathcal{I}\setminus\{I\},\ s\in\mathcal{S}, \tag{15}$$

$$B_i^{s+} = \max\{0, B_i^s\} \qquad i \in \mathcal{I}\setminus\{I\},\ s\in\mathcal{S}, \tag{16}$$

$$x_i \geq 0 \ \ integer \qquad i \in \mathcal{I}\setminus\{I\}, \tag{17}$$

$$y^s_{i,i+1}, I_i^{s+}, I_i^{s-}, B_i^{s+}, E_i^{s+}, a_i^s, b_i^s \geq 0 \ \ integer \qquad i \in \mathcal{I}\setminus\{I\},\ s\in\mathcal{S}, \tag{18}$$

$$I_i^s, B_i^s, E_i^s \ \ free\ and\ integer \qquad i \in \mathcal{I}\setminus\{I\},\ s\in\mathcal{S}. \tag{19}$$

The objective function (1) represents the minimization of the expected total cost obtained through the sum of penalties for the allocated bikes, the expected rebalancing cost between stations, the expected extra and excess penalties and expected stock-out penalty for shortage, plus the penalization of the slack variables. Constraints (2) encourage the target inventory quantity to be greater or equal than the maximum number of consecutive withdrawals, while constraints (3) encourage the number of free racks to be greater or equal than the maximum number of consecutive returns. To guarantee that a feasible solution can be found if the sum of the maximum number of consecutive withdrawn and returned bikes is greater than the station capacity (i.e., $g_i^s+h_i^s > Q_i$) the slack variables a_i^s and b_i^s allow for deviations from these quantities. The amount of such a deviation is determined by the model thanks to the penalization of these slack variables in the objective function. Constraints (4) guarantee that the sum between the quantity allocated and initially available at each station does not exceed the station capacity. Constraint (5) implies that the total number of delivered bikes to stations is less than the available quantity at the depot.

Constraints (6) ensure that the number of bikes carried by the vehicle during rebalancing never exceeds its capacity in each scenario $s \in \mathcal{S}$. We recall that rebalancing occurs on a fixed route that begins and ends at the depot, but that rebalancing does not involve bringing bikes from the depot to the first station on this route. Consequently, when determining the target inventory level of the first station, the model considers the impossibility of moving bikes from the depot to that station at a later stage through rebalancing, and a larger number of bikes will be allocated at the first station if needed.

Constraints (7) ensure that, for the depot, in each scenario $s \in \mathcal{S}$, the quantity at the end of the period is equal to the initial bike availability and the quantity received from the last visited station minus the quantities delivered to stations. Constraints (8) ensure that, in each scenario $s \in \mathcal{S}$, at the end of the rebalancing period, the number of bikes at the depot does not exceed its capacity. Moreover, the "flow balance" constraints for bikes at the first station on this route is different from the remaining stations. Specifically, constraints (9) ensure that, for the first visited station, the quantity at the end of rebalancing is equal to the sum between the initial available quantity and the quantity received from the depot minus the quantities used to satisfy the demand and those bikes that are redistributed to subsequent stations on the route in each scenario $s \in \mathcal{S}$. Similarly, constraints (10) determine the inventory position (which can be negative or positive) at a station other than the first, as a function of the initial inventory level, the number allocated, the number withdrawn/returned, and the number redistributed to another station in each scenario $s \in \mathcal{S}$.

Constraints (11) and (12) determine the surplus and stock-out quantities, respectively, for each station and for each scenario $s \in \mathcal{S}$. Constraints (13) and (14) calculate the number of bikes at each station that are in excess of station capacity, in each scenario $s \in \mathcal{S}$ and are collected thanks to valet service. Similarly, constraints (15) and (16) determine, for each scenario $s \in \mathcal{S}$, when there are more bikes positioned at a station after rebalancing than were initially allocated, and less or equal to station capacity.

Finally, constraints (17) to (19) define the domain of the variables.

5. Methodology

In this section, we describe the methods that we use for our analysis. In Section 5.1, we explain how scenarios have been generated and determine the size of the related scenario tree. The indicators used to analyze the value of rebalancing and of including rebalancing information are described in Section 5.2. In Section 5.3, we present the indicators that we use to analyze the stochastic solution. In Section 5.4, we describe the matheuristic we propose. Finally, in Section 5.5, we describe the simulation model used to compare the performance of target inventory levels suggested by different approaches.

5.1. Scenario generation method

Scenario generation is an important part of the modeling process, since a bad scenario tree can lead to a solution of the optimization problem that is not meaningful. Consequently, in this section, we aim at understanding how to generate realistic scenarios.

We recall that a typical assumption of stochastic programming is that the distribution of the random variable is known. However, in most practical applications, the distributions of the stochastic parameters have to be approximated by discrete distributions with a limited number of outcomes. The discretization is called a scenario tree. We assume that the random variables, which are the demand and the maximum number of consecutive withdrawals and returns at each station, have a finite number of possible outcomes, assumed to be exogenous to the problem. Consequently, the probability distribution is not influenced by decisions. Making these assumptions, we can represent the stochastic demand d_i^s, $i \in \mathcal{I}$, $s \in \mathcal{S}$, the maximum number of consecutive withdrawals g_i^s, $i \in \mathcal{I}$, $s \in \mathcal{S}$, and the maximum number of consecutive returns h_i^s, $i \in \mathcal{I}$, $s \in \mathcal{S}$, using a scenario tree which contains a root and a finite set of leaves.

Our method for generating scenarios presumes ridership data consisting of all withdrawal and return events occurred at each station i on N days. For each of such events, the time at which a bike has been withdrawn and returned is also available. We extract, for each station $i \in \mathcal{I} \setminus \{I\}$ and day $n \le N$, the sequence of all withdrawal and return events within the considered time interval, and we denote this sequence for station i on day n by ϕ_i^n. Note that we do not consider censored demand (i.e., invisible demand for withdrawals and returns that cannot be observed because the station has no bikes or free docks) due to its unavailability from the ridership data. For censored demand estimation, the interested reader can refer to approaches based on data preprocessing techniques (Albiński et al., 2018), simulation (Negahban, 2019), or predictive models (Gammelli et al., 2020). From the sequence ϕ_i^n, we compute:

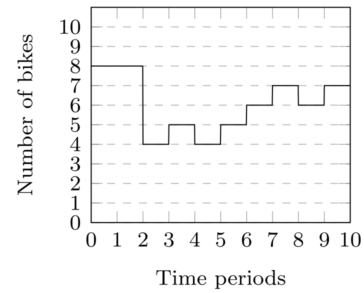


Fig. 2. Example of the evolution of the number of bikes for a given time interval at a given station.

- The net demand as $d_i^n = w_i^n - s_i^n$, where w_i^n and s_i^n represent the total number of withdrawn and returned bikes, respectively, at station i on day n in the considered time interval. Since we add penalties in the objective function for excess inventory and for stockout, we do not bound the demand values to stations capacities.
- The maximum number g_i^n of consecutive withdrawn bikes before a return occurs.
- The maximum number h_i^n of consecutive returned bikes before a withdrawal occurs.

This way, we obtain, for each station and day, a quadruplet $(\phi_i^n, d_i^n, g_i^n, h_i^n)$. Fig. 2 shows an example of the evolution of the number of bikes at a station for ten time periods. In this example, the sequence of withdrawal and return events is represented by $\phi_i^n = \{-4, 1, -1, 1, 1, 1, -1, 1\}$. Because there are 6 withdrawn and 5 returned bikes, the net demand is obtained as $d_i^n = 6 - 5 = 1$. The maximum number of consecutive withdrawals before a return occurs corresponds to $g_i^n = 4$ (see time period 2), and the maximum number of consecutive returns before a withdrawal occurs is $h_i^n = 3$ (see time periods from 5 to 7).

Finally, for each station $i \in \mathcal{I} \setminus \{I\}$, we derive the empirical distribution of its quadruplets $(\phi_i^n, d_i^n, g_i^n, h_i^n)$ as inverse of the Kaplan–Meier estimate of the cumulative distribution function (also known as the empirical cdf). Using this empirical cdf, scenario trees of any size can be generated according to a Monte Carlo sampling procedure. Because the empirical distribution is quadruplet-based, the four stochastic parameters have a joint distribution. In the sampling process, Pseudo-code (1) in Appendix C presents the details for scenario generation.

5.1.1. Determining the size of the scenario tree

In this section, we describe the procedure to determine the size of the scenario tree to accurately describe the stochastic parameters. To determine the minimum number of scenarios to consider, we perform both an in-sample and out-of-sample stability analysis of our stochastic program, following the procedure described in Kaut and Wallace (2007).

Regarding the in-sample analysis, we solve an SP for scenario trees of increasing size. Specifically, we generate one scenario tree for each size. For each scenario tree size, we store the obtained SP first-stage variable solutions.

Because these in-sample objective values are not directly comparable (they are computed by using scenario trees of different sizes), an out-of-sample stability analysis is necessary to compute the out-of-sample costs to estimate the effect of using a larger scenario tree. For this purpose, we declare a large scenario tree to be the true representation of the real world, and we use it as a benchmark to evaluate the cost of the optimal solutions obtained from the in-sample analysis, i.e., using scenario trees with a smaller size. The smaller scenario tree size at which the objective function value becomes stable (i.e., stops varying by a large amount) determines the number of scenarios to consider.

5.2. Indicators for analyzing the value of rebalancing and of anticipating rebalancing

In this section, we present the indicators used to assess the value of rebalancing and the value of anticipating rebalancing.

With the *Value of Rebalancing* (VR), we measure the benefits of rebalancing in comparison to a bike-sharing system that does not do it. To compute this indicator, we denote the objective function value of the SP by RP. Moreover, we consider the optimal objective function value $\mathrm{RP}^{\mathrm{wo\_reb}}$ obtained by solving the $\mathrm{SP}^{\mathrm{wo\_reb}}$, that is the SP in which rebalancing is forbidden (i.e., all rebalancing variables and parameters are removed from the model). Then, we compute the (percentage) VR as:

$$\mathrm{VR}(\%) = [(\mathrm{RP}^{\mathrm{wo\_reb}} - \mathrm{RP})/\mathrm{RP}] \times 100.$$

The *Value of Anticipating Rebalancing* (VAR) measures the advantage of considering rebalancing when the target inventory levels are determined compared to ignoring it. For this, we evaluate how the $\mathrm{SP}^{\mathrm{wo\_reb}}$ solution performs in a bike-sharing system with rebalancing by computing the $\mathrm{RP}^{\mathrm{reb}}$. This objective function value $\mathrm{RP}^{\mathrm{reb}}$ is obtained by fixing the $\mathrm{SP}^{\mathrm{wo\_reb}}$ first-stage decisions in the SP. We compute the (percentage) VAR as:

$$\mathrm{VAR}(\%) = [(\mathrm{RP}^{\mathrm{reb}} - \mathrm{RP})/\mathrm{RP}] \times 100.$$

5.3. Indicators for the analysis of stochastic solution

In this section, we present the indicators adopted to analyze the value of stochastic solution (Section 5.3.1), and the quality of the expected value solution (Section 5.3.2).

5.3.1. Indicators for analyzing the value of the stochastic solution

To determine the importance of including uncertainty in our model, we compute the *Value of Stochastic Solution* (VSS) (Birge and Louveaux, 2011; Kall and Wallace, 1994).

First, we solve the SP on the scenario tree generated by applying the methods described in Section 5.1 to get the optimal SP objective function value, that we call RP. We then solve the *Expected Value problem* (EVP) and get its optimal objective function value EV. The EVP is the deterministic version of our SP obtained by replacing the random parameters for demand, and maximum consecutive number of withdrawn and return bikes with their expected values rounded to the nearest integer (these quantities cannot be represented by fractional values). We then evaluate how the EVP solution performs in the stochastic setting by computing the *Expectation of Expected Value* (EEV), obtained by fixing the first-stage expected value decisions in the SP. Finally, we compute the (percentage) VSS as:

$$\mathrm{VSS}(\%) = [(\mathrm{EEV} - \mathrm{RP})/\mathrm{RP}] \times 100.$$

5.3.2. Indicator of the structure of the expected value solution

To assess the quality of the EVP solution when solving the SP, we compute the *Loss of Using the Skeleton Solution* (LUSS) (Maggioni and Wallace, 2012).

For our problem, we interpret the skeleton solution as the stations to which no bike is allocated. To compute the LUSS, we examine the solution of the EVP to determine the subset of stations $\bar{\mathcal{I}}$ to which it allocates zero bikes. We then solve the SP, fixing $x_i = 0$ for stations $i \in \bar{\mathcal{I}}$. We refer to the objective function value of the optimal solution to this problem as the *Expected Skeleton Solution Value* (ESSV) and compute the (percentage) LUSS as:

$$\mathrm{LUSS}(\%) = [(\mathrm{ESSV} - \mathrm{RP})/\mathrm{RP}] \times 100.$$

5.4. A matheuristic algorithm based on the upgradability of the expected value solution

To assess the upgradability of the EVP solution when solving the SP, we compute the *Loss of Upgrading the Deterministic Solution* (LUDS) (Maggioni and Wallace, 2012). With the LUDS, we seek to determine whether the solution to the EVP is upgradeable, i.e., it can be used as a starting point for generating a high-quality solution to the SP. To do so, we solve the SP, albeit with additional constraints ensuring that the values of the first-stage variables are at least as large as their values in the optimal solution to the EVP. We refer to the objective function value of the optimal solution to this restricted SP as the *Expected Input Value* (EIV) and compute the (percentage) LUDS as:

$$\mathrm{LUDS}(\%) = [(\mathrm{EIV} - \mathrm{RP})/\mathrm{RP}] \times 100.$$

The procedure to compute the LUDS can be viewed as a matheuristic in which a mathematical model, based on information from the deterministic solution, is solved. Our aim is to show that this matheuristic performs better than the deterministic solution on which it is based, and that its running time is smaller than the one needed to optimally solve the problem.

5.5. Simulation model

To simulate the performance of an initial allocation of bikes to stations, we consider the historical ridership data of a week.

For each day during this week, we input the number of allocated bikes to every station and simulate the movement of bikes based on rides taken on that day. We record statistics related to: (i) the number of times a user wants to return a bike to a station but it is full ("cong.(%)"), and (ii) the number of times a user wants to withdraw a bike from a station but it is empty ("starv.(%)"). Then, we compute the inventory level at each station before rebalancing is performed as $I_i^{\text{before rebalancing}} = \bar{I}_{i0} + x_i - w_i + s_i$, where w_i and s_i stand for the number of withdrawn and returned bikes, respectively, at station i, defined in the interval of integer numbers $[0, Q_i]$. From this, we solve the second stage of our SP to determine the number of rebalanced bikes ("#rebal."), and the final inventory level at each station after rebalancing is performed, i.e., $I_i^{\text{after rebalancing}} = I_i^{\text{before rebalancing}} + y_{i-1,i} - y_{i,i+1}$.

To measure the number of bikes that are not redistributed through rebalancing (it is not convenient according to our SP), by using $I_i^{\text{after rebalancing}}$, we compute the sum of the extra inventory balance over all stations ("B^+") that represents the total number of bikes in excess with respect to the sum of the target inventory levels of stations. Finally, to observe how the system performs after rebalancing has been carried out, we adopt an alternative perspective on the final inventory levels, and we calculate a fill rate-type statistic that estimates the percentage of future bike withdrawals that can be satisfied. For this, we use the same demand scenario set determined according to Section 5.1.1, and compute the fill rate of station i in scenario s as follows:

$$FR_i^s = \begin{cases} 1 & \text{if } I_i^{\text{after rebalancing}} - d_i^s \geq 0, \\ \frac{I_i^{\text{after rebalancing}}}{d_i^s} & \text{otherwise.} \end{cases} \tag{20}$$

To get the average expected fill rate, we average this statistic over all stations ($\overline{FR}(\%)$) and scenarios ($\mathbb{E}$) and, in the following, we refer to it with "$\mathbb{E}[\overline{FR}(\%)]$".

6. Numerical results

In this section, we present and analyze the results of our computational study. We describe the generated instances and scenarios in Section 6.1. Section 6.2 presents an assessment of the influence of the inclusion of the stochastic parameters g_i^s and h_i^s in SP (introduced to correct the static demand assumption). In Section 6.3, we assess the value of rebalancing and rebalancing information. Section 6.4 presents the results of the stochastic solution analysis, including the evaluation of the performance of the matheuristic we consider. We compare the target inventory levels determined by solving the SP to the real allocation plan implemented in the bike-sharing system of San Francisco in Section 6.5. Finally, in Section 6.6, we conclude with a sensitivity analysis to evaluate the behavior of the SP first-stage and second-stage cost for increasing values of the penalties in the SP second stage.

All computational experiments were run on a computer with 8 GB of RAM and a 2.70 GHz CPU. All software was implemented in Python 3.6.1, with optimization problems solved to optimality with Gurobi 9.5.2. All Gurobi parameters were set to their default values.

6.1. Instances and scenario tree

In order to study the impact of different numbers of stations, different lengths of the time interval, and different congestion and starvation penalties on the solutions of our SP, we generate 30 instances based on the bike-sharing system of the city of San Francisco. The San Francisco bike-sharing system consists of 340 stations. Their location and the ridership data are available at the website https://www.lyft.com/bikes/bay-wheels/system-data (last accessed in February 2024).

In all instances, the depot capacity and initial bike availability $\overline{I}_{I0}$ is 7000 bikes, the initial bike availability at each station, $\overline{I}_{i0}, i \in \mathcal{I} \setminus \{I\}$ is 0 bikes, the station capacities Q_i are set equal to the ones available on the website, and the capacity of the vehicle used for rebalancing C is 25 bikes (see Forma et al. (2015)). Because the road network in San Francisco consists of blocks, all distances δ_{ij} between station i and j are calculated by using the Manhattan distance formula. Moreover, all instances share, for each station $i \in \mathcal{I} \setminus \{I\}$, the same allocation penalty $f_i = 1$, and the same rebalancing penalty $t_{i,i+1} = 2$.

To generate multiple instances, we consider all combinations of parameters summarized in Table 2. Specifically, we generate five bike-sharing systems, each having a different number of stations. For each bike-sharing system size, we randomly draw the specified number of stations from the total set of 340 stations. The bike-sharing system with 340 stations corresponds to the real one. Moreover, we consider three different time intervals. The lower bound of each interval is the time at which the target inventory level must be in place, while the upper bound indicates the time at which we assume rebalancing is performed for our cost estimation. Note that, even though we considered these time intervals, our approach can also be applied with longer time intervals (e.g., one day). Starvation is penalized by the term p_i modeling that when a user cannot withdraw a bike from the station she desire, she will walk to another station. Specifically, we consider two cases. The first case assumes that the user walks to the closest station and finds a bike resulting in $p_i = \kappa(\min_{j\in\mathcal{I}\setminus I: j\neq i} \delta_{ij})$. The second case assumes that an undefined number of neighboring stations are all short of bikes simultaneously and the user walks to the first-encountered stations containing an available bike resulting in $p_i = \kappa(\sum_{j\in\mathcal{I}\setminus I: j\neq i} \delta_{ij}/(|\mathcal{I}|-2))$, where $\sum_{j\in\mathcal{I}\setminus I: j\neq i} \delta_{ij}/(|\mathcal{I}|-2)$ represents the average distance of station i to any other station. In both cases, to make the penalties of the same magnitude of the allocation and rebalancing penalties, we set $\kappa = 1000$. To penalize congestion and starvation in the same way, we set $c_i = p_i$. In fact, during peak hours, it can happen that the valet at a station has to handle too many bikes resulting in users who have to wait for getting a free dock and, therefore, decide to travel to other stations. By considering all combinations of parameters, we obtain 30 instances.

Table 2
Instance parameter combinations.

Parameter	Values
Number of stations $\lvert\mathcal{I}\rvert - 1$	{50, 100, 200, 300, 340}
Time interval	{[6:00–8:00], [6:00–10:00], [6:00–12:00]}
Excess and stockout penalties c_i, p_i	$\{\kappa(min_{j\in\mathcal{I}\setminus I: j\neq i}\delta_{ij}), \kappa(\sum_{j\in\mathcal{I}\setminus I: j\neq i} \delta_{ij}/(\lvert\mathcal{I}\rvert - 2))\}$

6.1.1. Scenario generation and size of the scenario tree

To generate scenarios, we follow the approach described in Section 5.1. Specifically, we use the real historical ridership data (https://www.lyft.com/bikes/bay-wheels/system-data) of the months from May to August of the years 2022 and 2023, and the following time intervals depending on the instance under consideration: from 6:00 a.m. until 8:00 a.m., from 6:00 a.m. until 10 a.m., and from 6:00 a.m. until 12 p.m.

To determine how many scenarios to consider to accurately describe the stochastic demand, we then perform an in-sample and an out-of-sample stability analysis. Table 3 summarizes the results obtained for the in-sample stability analysis. For each scenario tree size, we report the objective function value and the runtime (in seconds), both averaged over all 30 instances. The results (also represented in Fig. 3(a)) show that the average objective function value stabilizes with 500 scenarios.

For the out-of-sample stability analysis, we declare a scenario tree with 700 scenarios to be the true representation of the real world, and we use it as a benchmark. Fig. 3(b) shows, for each scenario tree size, the gap of the objective function value obtained with that scenario tree size to the objective function value obtained using the benchmark scenario tree averaged over all instances ($\overline{\Delta}(\%)$). The results show that the convergence of the out-of-sample analysis is monotonic and decreasing, and that by considering a scenario tree size of 500, the average gap is below 1%.

Because considering a high number of scenarios increases the runtimes, it is recommendable to select the smallest scenario tree size at which the objective function value shows convergence to the one obtained by using the benchmark scenario tree. Hence, in the following, we base our computational study on a set of 500 scenarios which we assume to be equiprobable, i.e., we set $pr^s = 1/|\mathcal{S}|, s \in \mathcal{S}$.

6.2. Assessing the influence of the inclusion of the stochastic parameters g_i^s and h_i^s in SP

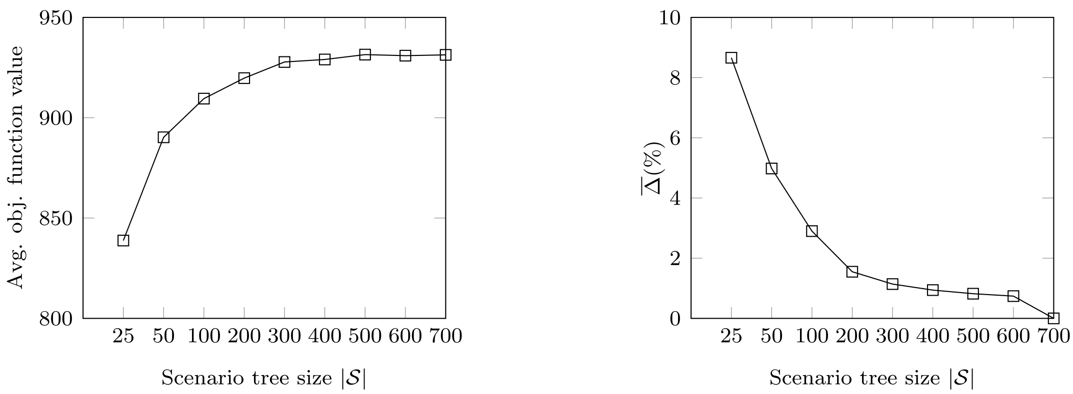
In this section, we compare the solution of the SP to the solution obtained by ignoring the penalization of the parameters g_i^s and h_i^s in the objective function and constraints (2) and (3). We refer to this new formulation with SP$^{\text{mod}}$. For this comparison, we assess how well the target inventory levels suggested by the SP$^{\text{mod}}$ and the SP solutions perform in the simulation model introduced in Section 5.5.

Table 8 in Appendix D.1 reports the detailed results, while Table 4 shows the average simulation model statistics related to the performance of each target inventory level plan. In the SP, the target inventory levels are 130.98% on average higher than the ones suggested by the SP$^{\text{mod}}$ solution. Hence, in the SP, the frequency of starvation is much smaller and the expected fill rates are higher. Due to the smaller number of bikes delivered in the SP$^{\text{mod}}$ solution, its frequency of congestion and the number of rebalanced bikes are smaller than the ones in the SP solution. However, the values of these two statistics are very small for the SP solution compared to the much higher frequency of starvation of the SP$^{\text{mod}}$ solution. Moreover, even though the target inventory levels are higher in the SP solution, the extra inventory levels are lower than in the SP$^{\text{mod}}$ solution. This can be explained by a better allocation of bikes to stations that leads to more effective rebalancing activities.

Hence, the consideration of the parameters g_i^s and h_i^s in SP leads to an improved solution compared to the solution returned by ignoring them.

Table 3
Average results of the in-sample stability analysis computed over 30 instances.

	Scenario tree size $\lvert\mathcal{S}\rvert$								
	25	50	100	200	300	400	500	600	700
Average objective function value	838.80	890.26	909.53	919.75	927.79	929.04	931.47	930.94	931.34
Average runtime (seconds)	5.30	11.78	34.27	83.75	146.32	216.97	295.64	494.95	546.43



(a) In-sample stability analysis.

(b) Out-of-sample stability analysis.

Fig. 3. Average results of the in sample and out-of-sample stability analysis computed over 30 instances.

Table 4
Average results for the simulation-based comparison of solutions of the SP and SP^{mod} problems computed over 30 instances.

	SP						SP^{mod}					
	$\sum_{i\in\mathcal{I}\setminus\{I\}} x_i$	*cong.*(%)	*starv.*(%)	#rebal.	B^+	$\mathbb{E}[FR(\%)]$	$\sum_{i\in\mathcal{I}\setminus\{I\}} x_i$	*cong.*(%)	*starv.*(%)	#rebal.	B^+	$\mathbb{E}[FR(\%)]$
Avg.	594.83	0.12	6.11	5.78	177.26	94.71	257.53	0.07	25.42	4.40	200.46	88.47

6.3. The value of rebalancing and of anticipating rebalancing

In this section, we assess the value of rebalancing (VR) and of anticipating rebalancing (VAR) introduced in Section 5.2.

For the VR, the detailed results by instance can be found in Table 9, while Fig. 4 shows the VR(%) results aggregated by number of stations, time interval, and penalty. The average VR(%) is equal to 5.96% showing that bike-sharing systems that implement rebalancing achieve a better performance compared to systems that do not execute it. This is especially true for instances in which the penalty for congestion and starvation at a station is set equal to the average distance to any other station (see Fig. 4(c)) and for instances in which the length of the time interval between the moment in which the target inventory level is applied and rebalancing is performed is longer (see Fig. 4(b)). In the first case, rebalancing is more beneficial because penalties are higher compared to the minimum distance case. In the second case, with longer intervals, the system is more unbalanced, and hence, the value that rebalancing can bring to the system is higher.

The detailed results by instance of the VAR can be found in Table 10. The average VAR(%) is equal to 1.29% with a standard deviation of 1.13%, suggesting that, when determining the target inventory level, the consideration that rebalancing will be performed at a certain point in time is beneficial. The value of anticipating rebalancing is higher for those instances in which the length of the time horizon between the moment in which the target inventory level is applied and rebalancing is performed is longer (see Fig. 5(b)), and in which the penalty for congestion and starvation at a station is set equal to the average distance to any other station (see Fig. 5(c)).

6.4. Stochastic solution analysis and evaluation of the matheuristic

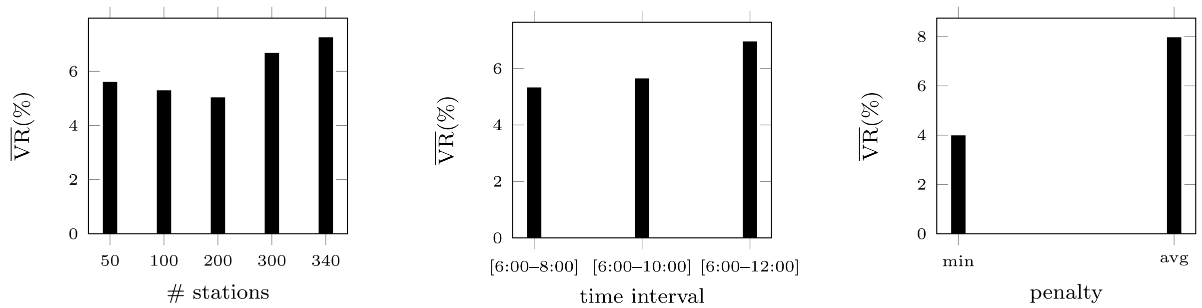
In this section, we analyze the value of the SP solution (Section 6.4.1), and the quality of the EVP solution, including the evaluation of the performance of the matheuristic (Section 6.4.2).

6.4.1. Analyzing the value of the stochastic solution

In this section, we assess the value of the stochastic solution (VSS).

The detailed results by instance can be found in Table 11 in Appendix D, while Fig. 6 shows the VSS(%) results aggregated by number of stations, time interval, and penalty. The average $\overline{\text{VSS}}(\%)$ is equal to 185.50% suggesting that significant gains can be obtained by solving the SP instead of using the expected value approach. Figs. 6(a) and 6(b) show that as the number of stations and the length of the time interval increase, the gains of solving the SP instead of using the deterministic solution decrease but stay relevant. Moreover, Fig. 6(c) shows that by considering the average distance of a station to all other stations as penalty for starvation and congestion, the $\overline{\text{VSS}}(\%)$ is considerably higher than considering the distance to its closest station. The results suggest that including uncertainty in the demand, and in the number of maximum consecutive, and returned bikes is always relevant, especially in bike-sharing systems that consider shorter time intervals between the moment in which the target inventory levels are implemented and the moment in which rebalancing takes place, and that are characterized by clusters of neighboring stations performing starvation and/or congestion at the same time.

We also assess how well the initial allocation plans from the EV solution and the SP solution perform in the simulation model presented in Section 5.5. Table 12 in Appendix D reports the detailed results, while Table 5 shows the average simulation model statistics related to the performance of each plan. The SP allocates on average 132.38% more bikes, leading to a much smaller frequency of starvation, and higher fill rates. Due to the smaller number of bikes delivered in the EVP solution, its frequency of congestion and number of rebalanced bikes are smaller than the SP solution. However, the values of both these statistics are very limited for the SP solution compared to the higher frequency of starvation of the EVP solution. Interestingly, despite of the higher number of bikes delivered in the SP solution, the extra inventory levels are lower than in the EVP solution. This may suggest a better

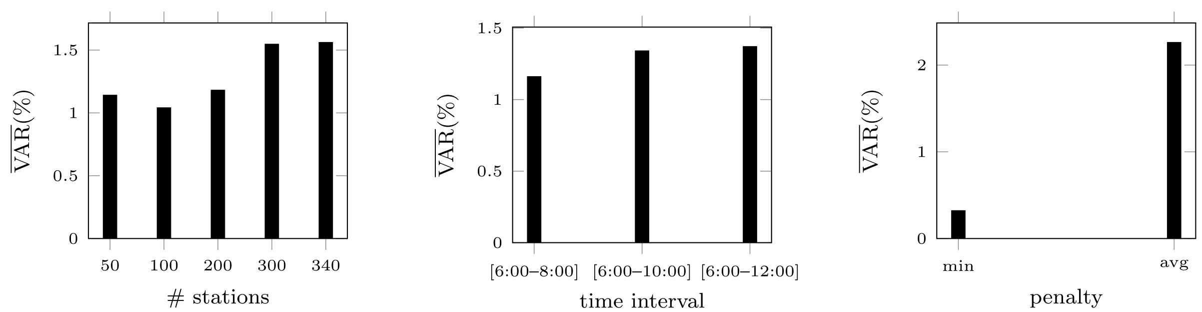


(a) VR(%) results aggregated by number of stations.

(b) VR(%) results aggregated by time interval.

(c) VR(%) results aggregated by penalty.

Fig. 4. VR (%) results aggregated by number of stations, time interval, and penalty.

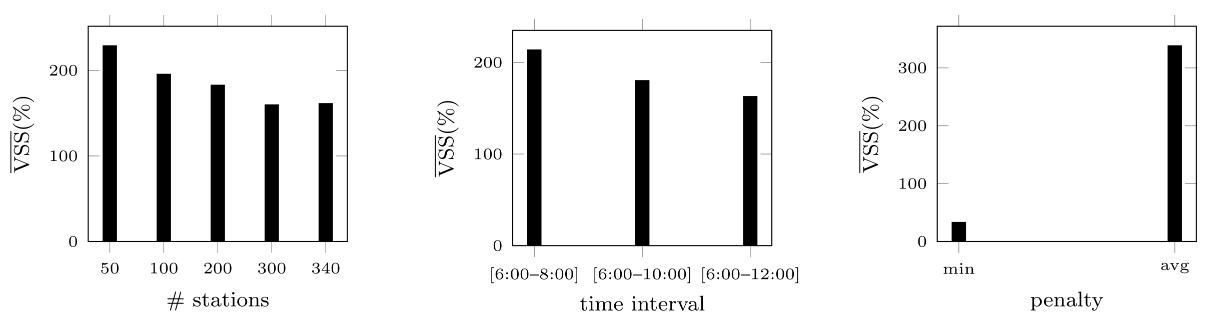


(a) VAR(%) results aggregated by number of stations.

(b) VAR(%) results aggregated by time interval.

(c) VAR(%) results aggregated by penalty.

Fig. 5. VAR (%) results aggregated by number of stations, time interval, and penalty.



(a) VSS(%) results aggregated by number of stations.

(b) VSS(%) results aggregated by time interval.

(c) VSS(%) results aggregated by penalty.

Fig. 6. VSS (%) results aggregated by number of stations, time interval, and penalty.

Table 5
Average results for the simulation-based comparison of solutions of the stochastic and expected value problems computed over 30 instances.

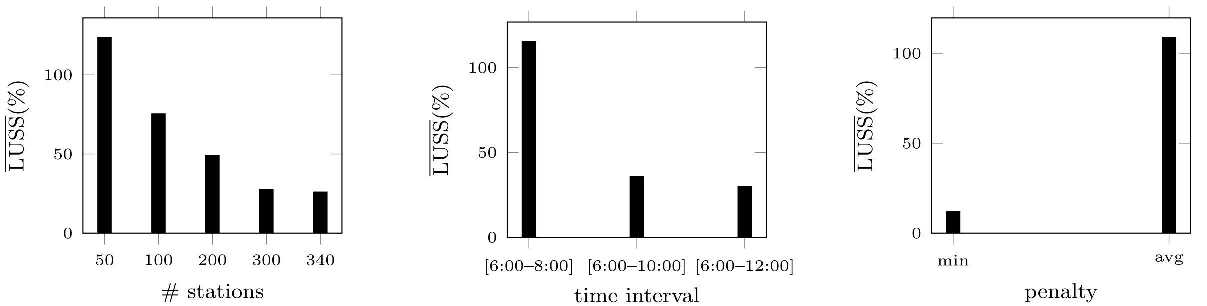
	SP						EV					
	$\sum_{i\in\mathcal{I}\setminus\{I\}} x_i$	*cong.*(%)	*starv.*(%)	#rebal.	B^+	$\mathbb{E}[FR(\%)]$	$\sum_{i\in\mathcal{I}\setminus\{I\}} x_i$	*cong.*(%)	*starv.*(%)	#rebal.	B^+	$\mathbb{E}[FR(\%)]$
Avg.	594.83	0.12	6.11	5.78	177.26	94.71	255.97	0.08	26.25	4.37	184.52	86.29

allocation of bikes to stations that leads to more effective rebalancing operations.

6.4.2. Analyzing the quality of the expected value solution and of the matheuristic algorithm

In this section, we assess the quality of the EVP solution when solving the stochastic model, and of the matheuristic algorithm.

The detailed results of the LUSS(%) can be found in Table 13 in Appendix D. The average $\overline{\text{LUSS}}(\%)$ is equal to 60.35%. This positive value means that the EVP solution selects to allocate no bikes to wrong stations and, consequently, its structure (skeleton) cannot be inherited in a stochastic environment. From Table 13, we observe that the EVP solution always does not allocate any bike to too many stations than the ones without bikes in the SP solution. We also notice that the

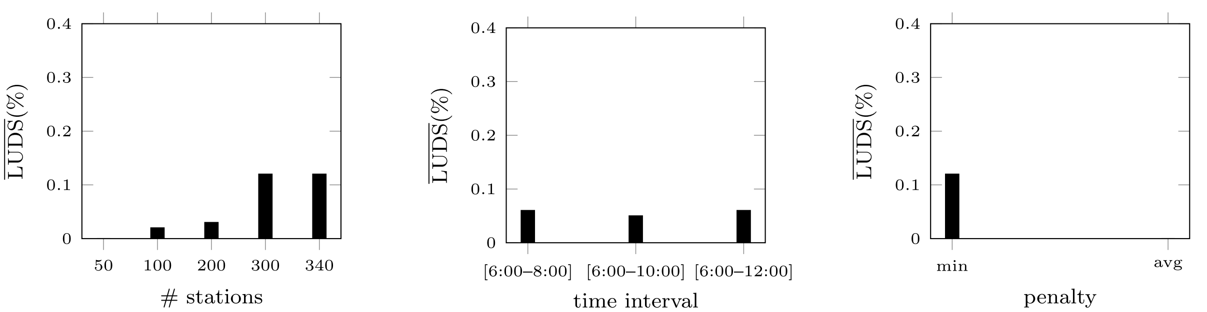


(a) LUSS(%) results aggregated by number of stations.

(b) LUSS(%) results aggregated by time interval.

(c) LUSS(%) results aggregated by penalty.

Fig. 7. LUSS (%) results aggregated by number of stations, time interval, and penalty.



(a) LUDS(%) results aggregated by number of stations.

(b) LUDS(%) results aggregated by time interval.

(c) LUDS(%) results aggregated by penalty.

Fig. 8. LUDS (%) results aggregated by number of stations, time interval, and penalty.

stations that do not receive any bikes in the SP solution are also not receiving any bikes in the EVP solution (but the opposite is not true). The LUSS (%) results aggregated by number of stations, time interval, and penalty and displayed in Fig. 7 show the same behavior observed for the VSS (%).

The detailed results of the LUDS (%) are reported in Table 14 in Appendix D. Recall that the value of this indicator, together with the computational times, allows us to evaluate the performance of the matheuristic. The average LUDS (%) is equal to 0.06%. In particular, it is equal to 0.00% for 20 out of 30 instances. These results suggest that the EVP solution is *perfectly upgradeable* in most of the instances, indicating that solving the EVP can be a good start for solving the SP. Even though, in the remaining ten instances, the EVP solution is only partially upgradeable, the maximum value of the LUDS (%) is 0.29%. This shows that a very good solution quality to the SP can be obtained by simply upgrading the EVP solution. Besides, by first solving the EVP, and then the EIV problem, the total solution time is reduced by almost 11%, on average, of what it is needed to solve the SP from scratch. This result shows that such a matheuristic procedure based on the use of information from the deterministic solution is valuable.

The aggregated results reported in Fig. 8 do not suggest any particular behavior of the LUDS(%) regarding the length of the time interval. Instead, the EVP solution seems to become less upgradeable as the number of stations increases. Moreover, solutions of the same quality as the optimal ones can be obtained in the instances in which the penalty is based on the average distance of a station to all other stations as suggested by the LUDS(%) which is equal to zero for these instances (see Fig. 8(c)). The reason can be due to the fact that, in these instances, the penalties for congestion and starvation are higher. Recalling that the penalization for congestion is prioritized over the one for starvation, in these instances, the EVP solution always suggests lower target inventory levels for all stations compared to the ones in the SP solution.

6.5. A comparison with the implemented system

In this section, we evaluate the performance of the actual allocation plan implemented in the bike-sharing system of San Francisco compared to the one obtained by solving our SP. For this, we use the simulation model described in Section 5.5, and consider the complete bike-sharing system consisting of 340 stations. Because accessing the past actual allocation of bikes to stations is not possible, we extract the data of the station status (i.e., the number of bikes at each station) at 6 a.m. of an exemplary working day (March 19, 2024). To normalize the comparison, we solve our SP under the constraint that the total number of allocated bikes has to correspond to the total allocated on that day in the real system. This allows us to assess the benefit of a different allocation of bikes to stations.

We present statistics regarding each plan in Table 6. With respect to the statistics presented in Section 5.5, we additionally report the sum over stations of the target inventory levels ("$\sum_{i\in\mathcal{I}\setminus\{I\}} x_i$"). Given the same total number of allocated bikes, we observe that the allocation plans prescribed by our SP outperform the actual allocation plans. In fact, the plans suggested by our SP lead to fewer congestion and starvation events, fewer rebalanced bikes, and higher expected fill rates. Only the extra inventory level is, on average, higher in the SP plan compared to the actual one, but only by a small amount. These results represent a strong indicator of the impact that the proposed stochastic approach could have in practice.

6.6. Sensitivity analysis for the penalty terms in the SP second stage

In all previous experiments, we used the values of the penalties described in Section 6.1. While delivery and rebalancing penalties can be easily quantified by observing tangible factors such as material and workforce costs, congestion and starvation penalties are more difficult

Table 6
Weekly-averaged results of the simulation model run for the SP solution and the real allocation system of the San Francisco bike-sharing system with 340 stations.

Instance			SP						Actual allocation plan					
# stations	Time interval	Penalty	$\sum_{i\in\mathcal{I}\setminus\{I\}} x_i$	*cong.*(%)	*starv.*(%)	#rebal.	B^+	$\mathbb{E}[FR(\%)]$	$\sum_{i\in\mathcal{I}\setminus\{I\}} x_i$	*cong.*(%)	*starv.*(%)	#rebal.	B^+	$\mathbb{E}[FR(\%)]$
340	[6:00–8:00]	min	3317.00	0.06	0.00	0.00	154.00	99.69	3317.00	1.54	3.11	3.00	155.71	98.61
340	[6:00–8:00]	avg	3317.00	0.00	0.00	11.43	143.00	99.70	3317.00	1.54	3.11	18.57	146.43	98.70
340	[6:00–10:00]	min	3317.00	0.19	0.11	0.00	393.43	99.08	3317.00	2.15	4.47	3.00	386.29	95.21
340	[6:00–10:00]	avg	3317.00	0.36	0.04	35.29	358.43	99.17	3317.00	2.15	4.47	35.14	362.43	95.37
340	[6:00–12:00]	min	3317.00	1.38	0.25	0.00	552.71	98.00	3317.00	3.77	5.72	3.00	535.86	93.24
340	[6:00–12:00]	avg	3317.00	1.59	0.27	38.29	513.14	98.09	3317.00	3.77	5.72	35.00	512.14	93.47
Avg.			3317.00	0.60	0.11	14.17	352.45	98.96	3317.00	2.48	4.43	16.29	349.81	95.77

to measure because tied to intangible elements such as customer satisfaction. In this section, we present how the first and the second-stage objective function values vary when only the congestion and starvation penalties increase, and when all penalty terms in the SP second stage increase.

Table 15 in Appendix D.4 reports the detailed results of the first and second-stage objective function values obtained for increasing values of stockout, excess, and extra inventory penalties. To obtain these results, we multiply stockout, excess, and extra inventory penalties by a constant value u, and solve our SP. In our experiments, we consider $u = \{1, 10, 100, 1000, 10000\}$, where $u = 1$ corresponds to the parameter values defined in Section 6.1. The results show that, for increasing values of the SP second-stage penalty terms, both the first and the second-stage objective function values tend to increase. Interestingly, we observe that for the instances considering a congestion and starvation penalty computed as the average distance to any other station, the first-stage cost stops changing starting from $u = 100$, while for the instances considering a congestion and starvation penalty computed based on the distance to the closest station, the first-stage cost stops changing from $u = 1000$. We notice that, for increasing values of u, some of the instances report not only the same first-stage cost but also the same second-stage cost. This occurs when, in the second-stage solution, there are no congestion or starvation quantities but only rebalancing ones. Because in this experiment, rebalancing penalties are not multiplied by u, the costs are the same.

Similar to Table 15, Table 16 in Appendix D.4 reports the detailed results of the first and second-stage objective function values albeit obtained for increasing values of all SP second-stage penalty terms (i.e., rebalancing, stockout, excess, and extra inventory penalties). For this, we multiply rebalancing, stock-out, excess, and extra inventory penalties by a constant value u, and solve our SP. Consistently to our previous experiments, we consider $u = \{1, 10, 100, 1000, 10000\}$, where $u = 1$ corresponds to the parameter values defined in Section 6.1. Also in this case, the results show that, for increasing values of the SP second-stage penalty terms, both the first and the second-stage objective function values tend to increase. After a certain u value, the first-stage cost becomes stable. However, in this case, it is not possible to derive any instance-based insight on the behavior of the first-stage cost. Instances whose solutions have the same first-stage cost for increasing values of u show different second-stage costs. However, we observe that, for these instances, the second-stage cost obtained for a higher value of u (e.g., $u = 10000$) exactly corresponds to the second-stage cost obtained for the lower value of u (e.g., $u = 1000$) multiplied by 10. In those cases, the value of the second-stage variables is the same only penalized by a bigger u value.

7. Conclusion and future work

In this paper, we studied the problem of determining the target inventory level of stations by considering the opportunity to perform rebalancing at a later point in time, in the context of a bike-sharing system. One of the challenges in determining these target inventory levels is that there are multiple dimensions along which the performance of such inventory levels can be measured. Some of them measure the costs incurred while operating the system. Others measure the quality of the service experienced by users of the system. As a result, we present a two-stage SP wherein the first-stage variables determine these target inventory levels, and the second-stage variables determine an approximation of the rebalancing cost by deciding how many bikes to rebalance at a point later in the day.

We performed a computational study based on a set of instances derived from historical ridership data of the bike-sharing system of the city of San Francisco. With this study, we first ascertained that considering rebalancing when the target inventory levels are determined is beneficial. Then, we established that, by not recognizing variability in bike station demand, the deterministic problem allocated too few bikes to too few stations. However, we also established that the time required to produce a high-quality solution to the stochastic program can be reduced by first solving its deterministic counterpart, providing in this way an effective matheuristic, based on the solution of the deterministic problem, to solve the stochastic problem. This finding can be valuable for practitioners, especially if this problem must be solved multiple times in a day. We also compared the performance of the target inventory levels prescribed by the SP with what we estimated was the actual allocation plan considering a given week of historical data. We saw that the SP produced a much better allocation of bikes than what is done in practice.

Regarding future work, we believe the next logical step is to consider a multi-stage SP that recognizes that rebalancing can occur multiple times throughout the day. This variant can be compared with the two-stage formulation provided in this paper, by means of rolling horizon approaches (see Bertazzi and Maggioni (2018)). Moreover, future works can be dedicated to adapt the SP formulation to solve, in limited runtime, the studied problem for instances having a larger amount of stations. Finally, interesting extensions could result by considering, e.g., the time at which rebalancing should be performed as a decision variable of the model, multiple vehicles available to perform rebalancing, and that their routes can change depending on stations' rebalancing needs.

CRediT authorship contribution statement

Rossana Cavagnini: Conceptualization, Data curation, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Francesca Maggioni:** Conceptualization, Investigation, Methodology, Supervision, Writing – original draft, Writing – review & editing. **Luca Bertazzi:** Conceptualization, Investigation, Methodology, Supervision, Writing – original draft, Writing – review & editing. **Mike Hewitt:** Conceptualization, Investigation, Methodology, Supervision, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Rossana Cavagnini reports financial support was provided by Transportation Science and Logistics Society. Francesca Maggioni reports financial support was provided by Francesco Severi National Institute of Higher Mathematics National Group of Scientific Calculations. Francesca Maggioni reports financial support was provided by ULTRA OPTYMAL, a PRIN 2020 project. If there are other authors, they 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 thank the Transportation and Logistics Society (TSL) as this research has been supported by the TSL Cross Region Doctoral Grant. This work has also been supported by "ULTRA OPTYMAL - Urban Logistics and sustainable TRAnsportation: OPtimization under uncertainTY and MAchine Learning", a PRIN2020 project funded by the Italian University and Research Ministry (grant number 20207C8T9M, official website: https://ultraoptymal.unibg.it) and by Gruppo Nazionale per il Calcolo Scientifico (GNCSINdAM). This study was carried out within the MOST–Sustainable Mobility National Research Center and received funding from the European Union Next-GenerationEU (PIANO NAZIONALE DI RIPRESA E RESILIENZA (PNRR)–MISSIONE 4 COMPONENTE 2, INVESTIMENTO 1.4–D.D. 1033 17/06/2022, CN00000023), Spoke 5 "Light Vehicle and Active Mobility". This manuscript reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them. Finally, the authors thank the Associate Editor and two anonymous Referees for providing valuable comments that helped to improve the paper.

Appendix A. Notation

See Table 7.

Appendix B. Model linearization

In this section, we describe the linearization of the model presented in Section 4. First, we replace the negative variable I_i^{s-} representing stock-out with the positive variable $I_i^{s,\text{stock-out}}$, and we and modify the objective function (1) as follows:

$$\min \sum_{i\in\mathcal{I}} f_i x_i + \sum_{s\in\mathcal{S}} pr^s [\sum_{i\in\mathcal{I}\setminus\{I\}} (t_{i,i+1} y^s_{i,i+1} + \frac{c_i}{Q_i} B_i^{s+} + c_i E_i^{s+} + p_i I_i^{s,\text{stock-out}} + p_i a_i^s + c_i b_i^s)]. \tag{21}$$

Then, we need to linearize the expressions for determining I_i^{s+} and $I_i^{s,\text{stock-out}}$. We introduce the binary variable z_i^s such that:

$$z_i^s = \begin{cases} 1 & \text{if } I_i^s \geq 0, \\ 0 & \text{otherwise.} \end{cases}$$

Table 7
Notation of the SP model.

Set	Description
$\mathcal{I} = \{1,\dots,I\}$	bike-stations (where I is the depot)
$\mathcal{S} = \{1,\dots,S\}$	scenarios or finite set of possible realization of the uncertainty
Deterministic parameter	**Description**
$\overline{I}_{I0}$	initial availability of bikes at the depot and depot capacity
Q_i	capacity of station $i \in \mathcal{I}\setminus\{I\}$
$\overline{I}_{i0}$	initial availability of bikes at station $i \in \mathcal{I}\setminus\{I\}$
C	capacity of the vehicle used for rebalancing
p_i	stock-out penalty at station $i \in \mathcal{I}\setminus\{I\}$
c_i	excess penalty at station $i \in \mathcal{I}\setminus\{I\}$
$\frac{c_i}{Q_i}$	extra penalty at station $i \in \mathcal{I}\setminus\{I\}$
f_i	penalty for each bike allocated at station $i \in \mathcal{I}\setminus\{I\}$
$t_{i,i+1}$	rebalancing penalty at station $i \in \mathcal{I}\setminus\{I\}$
Stochastic parameter	**Description**
d_i^s	demand of bikes at station $i \in \mathcal{I}\setminus\{I\}$ in scenario $s \in \mathcal{S}$
g_i^s	maximum number of consecutive bikes withdrawn from station $i \in \mathcal{I}\setminus\{I\}$ before a returnoccurs in scenario $s \in \mathcal{S}$
h_i^s	max. number of consecutive bikes returned to station $i \in \mathcal{I}\setminus\{I\}$ before a withdrawaloccurs in scenario $s \in \mathcal{S}$
pr^s	probability of scenario $s \in \mathcal{S}$
Variable	**Description**
$x_i \in \mathbb{Z}^+$	first-stage variable: number of bikes to ideally allocate at station $i \in \mathcal{I}\setminus\{I\}$
$y^s_{i,i+1} \in \mathbb{Z}^+$	second-stage variable: number of bikes relocated from station $i \in \mathcal{I}\setminus\{I\}$ to station $i+1$in scenario $s \in \mathcal{S}$
$I_i^s \in \mathbb{Z}$	second-stage variable: balance of bikes at station $i \in \mathcal{I}\setminus\{I\}$ in scenario $s \in \mathcal{S}$
$I_i^{s+} \in \mathbb{Z}^+$	second-stage variable: units of surplus at station $i \in \mathcal{I}\setminus\{I\}$ in scenario $s \in \mathcal{S}$
$I_i^{s-} \in \mathbb{Z}^-$	second-stage variable: units of stock-out at station $i \in \mathcal{I}\setminus\{I\}$ in scenario $s \in \mathcal{S}$
$B_i^s \in \mathbb{Z}$	second-stage variable: extra inventory balance at station $i \in \mathcal{I}\setminus\{I\}$ in scenario $s \in \mathcal{S}$
$B_i^{s+} \in \mathbb{Z}^+$	second-stage variable: units of extra inventory at station $i \in \mathcal{I}\setminus\{I\}$ in scenario $s \in \mathcal{S}$
$E_i^s \in \mathbb{Z}$	second-stage variable: excess inventory balance at station $i \in \mathcal{I}\setminus\{I\}$ in scenario $s \in \mathcal{S}$
$E_i^{s+} \in \mathbb{Z}^+$	second-stage variable: units of excess inventory at station $i \in \mathcal{I}\setminus\{I\}$ in scenario $s \in \mathcal{S}$
$a_i^s \in \mathbb{Z}^+$	second-stage variable: units of slack between the number of allocated bikes x_i and g_i^sat station $i \in \mathcal{I}\setminus\{I\}$ in scenario $s \in \mathcal{S}$
$b_i^s \in \mathbb{Z}^+$	second-stage variable: units of slack between the number of free racks $Q_i - x_i$ and h_i^sat station $i \in \mathcal{I}\setminus\{I\}$ in scenario $s \in \mathcal{S}$

and we substitute constraints (11) and (12) with:

$$I_i^{s+} \geq I_i^s \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S}, \tag{22}$$

$$I_i^{s+} \leq I_i^s + M(1 - z_i^s) \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S}, \tag{23}$$

$$I_i^{s+} \geq 0 \ \ integer \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S}, \tag{24}$$

$$I_i^{s+} \leq M z_i^s \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S}, \tag{25}$$

$$I_i^{s,\text{stock-out}} \geq -I_i^s \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S}, \tag{26}$$

$$I_i^{s,\text{stock-out}} \leq -I_i^s + M z_i^s \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S}, \tag{27}$$

$$I_i^{s,\text{stock-out}} \geq 0 \ \ integer \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S}, \tag{28}$$

$$I_i^{s,\text{stock-out}} \leq M(1 - z_i^s) \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S}. \tag{29}$$

The same linearization technique is applied for E_i^{s+}, and B_i^{s+}. In particular, we introduce the binary variables e_i^s and r_i^s, such that:

$$e_i^s = \begin{cases} 1 & \text{if } E_i^s \geq 0, \\ 0 & \text{otherwise.} \end{cases}$$

$$r_i^s = \begin{cases} 1 & \text{if } B_i^s \geq 0, \\ 0 & \text{otherwise.} \end{cases}$$

and we substitute constraints (14) and (16) with the following:

$$B_i^{s+} \geq B_i^s \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S}, \tag{30}$$

$$B_i^{s+} \leq B_i^s + M(1 - r_i^s) \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S}, \tag{31}$$

$$B_i^{s+} \leq M r_i^s \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S}, \tag{32}$$

$$B_i^{s+} \geq 0 \ \ integer \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S}, \tag{33}$$

$$B_i^{s-} \geq -B_i^s \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S}, \tag{34}$$

$$B_i^{s-} \leq -B_i^s + M r_i^s \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S}, \tag{35}$$

$$B_i^{s-} \leq M(1 - r_i^s) \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S}, \tag{36}$$

$$B_i^{s-} \geq 0 \ \ integer \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S}, \tag{37}$$

$$E_i^{s+} \geq E_i^s \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S}, \tag{38}$$

$$E_i^{s+} \leq E_i^s + M(1 - e_i^s) \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S}, \tag{39}$$

$$E_i^{s+} \leq M e_i^s \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S}, \tag{40}$$

$$E_i^{s+} \geq 0 \ \ integer \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S}, \tag{41}$$

$$E_i^{s-} \geq -E_i^s \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S}, \tag{42}$$

$$E_i^{s-} \leq -E_i^s + M e_i^s \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S}, \tag{43}$$

$$E_i^{s-} \leq M(1 - e_i^s) \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S}, \tag{44}$$

$$E_i^{s-} \geq 0 \ \ integer \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S}. \tag{45}$$

Finally, we introduce the variable definition constraints:

$$z_i^s \in \{0, 1\} \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S}, \tag{46}$$

$$r_i^s \in \{0, 1\} \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S}, \tag{47}$$

$$e_i^s \in \{0, 1\} \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S}. \tag{48}$$

Appendix C. Pseudo-code for the scenario generation process

In this section, Pseudo-code 1 presents the scenario generation process.

Pseudo-code 1: Scenario generation process.

Input: $(\phi_i^n, d_i^n, g_i^n, h_i^n),\ i \in \mathcal{I} \setminus \{I\},\ n = 1, \dots, N$
for $i \in \mathcal{I} \setminus \{I\}$ **do**
  $\mathcal{K} := \{n', n'' = 1, \dots, N : (\phi_i^{n'}, d_i^{n'}, g_i^{n'}, h_i^{n'}) \neq (\phi_i^{n''}, d_i^{n''}, g_i^{n''}, h_i^{n''})\}$
  for $k \in \mathcal{K}$ **do**
    $pdf_i[k] := \dfrac{\sum_{n',n''=1,\dots,N} \mathbb{1}_{((\phi_i^{n'}, d_i^{n'}, g_i^{n'}, h_i^{n'})=(\phi_i^{n''}, d_i^{n''}, g_i^{n''}, h_i^{n''}))}}{N}$
    $cdf_i[k] := pdf_i[k-1] + pdf[k]$
    $inv.cdf_i[k] := (\phi_i^k, d_i^k, g_i^k, h_i^k)$
  end
  for $s \in \mathcal{S}$ **do**
    sample a random number ("*random*") in [0,1]
    for $k \in \mathcal{K}$ **do**
      if $cdf_i[k-1] < random <= cdf_i[k]$ **then**
        $(\phi_i^s, d_i^s, g_i^s, h_i^s) = inv.cdf_i[k]$
      end
    end
    return d_i^s, g_i^s, h_i^s
  end
end

Appendix D. Detailed results

In this section, we report the detailed results for our experiments. Appendix D.1 shows the results detailed by instance of the experiments comparing formulation SP with formulation SP^{mod}. Appendix D.2 contains the results detailed by instance of the experiments assessing the value of rebalancing and of anticipating rebalancing. Appendix D.3 presents the detailed results by instance of the stochastic solution analysis. Finally, Appendix D.4 shows the results of the sensitivity analysis.

D.1. A comparison with the SP without the parameters g_i^s and h_i^s

See Table 8.

D.2. The value of rebalancing and of anticipating rebalancing

See Tables 9 and 10.

D.3. Stochastic solution analysis

See Tables 11–14.

D.4. Sensitivity analysis

See Tables 15 and 16.

Table 8
Detailed results for the simulation model run for the SP and the SP^{mod} solution.

Instance			SP						SP^{mod}					
#stations	Time interval	penalty	$\sum_{i\in I\setminus\{l\}} x_i$	*congestion*(%)	*starvation*(%)	rebal.bikes	B^+	$\mathbb{E}[FR(\%)]$	$\sum_{i\in I\setminus\{l\}} x_i$	*congestion*(%)	*starvation*(%)	rebal.bikes	B^+	$\mathbb{E}[FR(\%)]$
50	[6:00–8:00]	min	27.00	0.00	13.62	0.00	6.00	96.75	10.00	0.00	50.58	0.00	6.43	94.09
50	[6:00–8:00]	avg	65.00	0.00	0.00	0.43	5.14	99.47	21.00	0.00	19.19	0.57	5.71	96.00
50	[6:00–10:00]	min	60.00	0.00	6.06	0.00	13.43	95.64	26.00	0.00	26.55	0.00	15.57	91.15
50	[6:00–10:00]	avg	115.00	0.00	0.32	0.71	12.71	98.88	38.00	0.00	18.49	0.71	14.14	92.92
50	[6:00–12:00]	min	71.00	0.00	9.43	0.00	20.43	94.97	31.00	0.00	31.15	0.00	24.43	90.35
50	[6:00–12:00]	avg	133.00	0.00	1.30	1.29	18.14	98.49	45.00	0.00	24.42	1.14	22.00	92.72
100	[6:00–8:00]	min	65.00	0.00	18.49	0.00	18.29	95.54	30.00	0.00	42.46	0.00	19.43	92.23
100	[6:00–8:00]	avg	164.00	0.00	2.57	0.57	16.71	99.11	55.00	0.00	20.70	0.57	18.00	94.84
100	[6:00–10:00]	min	155.00	0.00	9.63	0.00	46.00	94.08	74.00	0.00	26.96	0.00	50.43	88.92
100	[6:00–10:00]	avg	301.00	0.00	1.22	3.57	42.00	98.79	107.00	0.00	20.80	2.86	46.14	91.96
100	[6:00–12:00]	min	206.00	0.00	9.51	0.00	72.29	93.10	94.00	0.00	27.14	0.00	81.86	87.38
100	[6:00–12:00]	avg	368.00	0.00	1.92	6.00	64.57	97.74	135.00	0.00	19.51	4.14	73.57	90.62
200	[6:00–8:00]	min	165.00	0.00	17.27	0.00	62.00	93.41	66.00	0.00	46.54	0.00	68.57	88.81
200	[6:00–8:00]	avg	399.00	0.00	2.02	2.71	56.14	98.81	137.00	0.00	24.63	1.86	63.43	92.61
200	[6:00–10:00]	min	370.00	0.00	8.28	0.00	147.29	91.26	177.00	0.00	27.98	0.00	165.14	85.35
200	[6:00–10:00]	avg	734.00	0.00	1.26	9.71	136.57	98.03	277.00	0.00	18.54	7.43	152.86	89.71
200	[6:00–12:00]	min	477.00	0.00	10.07	0.00	219.29	89.54	222.00	0.00	27.02	0.00	253.14	83.68
200	[6:00–12:00]	avg	906.00	0.00	2.15	14.14	202.29	96.83	345.00	0.00	17.88	9.57	228.29	88.01
300	[6:00–8:00]	min	365.00	0.00	13.82	0.00	149.29	91.14	152.00	0.00	40.39	0.00	167.57	85.09
300	[6:00–8:00]	avg	834.00	0.00	1.01	8.57	135.43	98.25	297.00	0.00	22.41	6.00	152.14	90.23
300	[6:00–10:00]	min	787.00	0.00	8.25	0.00	363.57	88.30	407.00	0.00	24.24	0.00	407.57	81.75
300	[6:00–10:00]	avg	1483.00	0.06	0.93	18.43	338.29	96.20	624.00	0.00	14.99	14.29	378.29	86.96
300	[6:00–12:00]	min	974.00	0.48	9.14	0.00	525.14	85.94	504.00	0.44	22.22	0.00	598.71	79.87
300	[6:00–12:00]	avg	1785.00	0.88	2.15	25.57	484.57	94.44	762.00	0.44	14.45	18.43	549.57	84.90
340	[6:00–8:00]	min	409.00	0.00	12.71	0.00	159.57	91.82	164.00	0.00	40.01	0.00	179.86	85.65
340	[6:00–8:00]	avg	928.00	0.00	0.91	11.29	143.43	98.50	338.00	0.00	20.55	8.57	160.71	91.08
340	[6:00–10:00]	min	869.00	0.03	7.77	0.00	402.29	88.67	459.00	0.00	23.26	0.00	451.14	82.38
340	[6:00–10:00]	avg	1615.00	0.18	0.94	33.43	362.29	96.55	703.00	0.00	14.47	27.71	408.71	88.05
340	[6:00–12:00]	min	1073.00	0.65	8.62	0.00	573.71	86.32	569.00	0.55	21.30	0.00	652.86	80.87
340	[6:00–12:00]	avg	1942.00	1.24	1.93	37.00	520.86	94.62	857.00	0.55	13.72	28.29	597.57	85.96
Avg.			594.83	0.12	6.11	5.78	177.26	94.71	257.53	0.07	25.42	4.40	200.46	88.47

Table 9
Detailed results for the VR(%) indicator.

Instance			RP	RP^{wo_reb}	VR(%)
#stations	Time interval	Penalty			
50	[6:00–8:00]	min	46.41	49.00	5.60
50	[6:00–8:00]	avg	95.19	101.14	6.25
50	[6:00–10:00]	min	88.77	93.34	5.15
50	[6:00–10:00]	avg	168.66	176.92	4.89
50	[6:00–12:00]	min	107.11	113.69	6.14
50	[6:00–12:00]	avg	195.00	205.77	5.52
100	[6:00–8:00]	min	111.06	117.42	5.72
100	[6:00–8:00]	avg	236.84	251.15	6.04
100	[6:00–10:00]	min	228.04	238.72	4.69
100	[6:00–10:00]	avg	448.09	472.55	5.46
100	[6:00–12:00]	min	287.46	299.68	4.25
100	[6:00–12:00]	avg	551.81	582.28	5.52
200	[6:00–8:00]	min	265.03	274.47	3.56
200	[6:00–8:00]	avg	603.66	640.90	6.17
200	[6:00–10:00]	min	540.03	558.63	3.44
200	[6:00–10:00]	avg	1148.49	1219.57	6.19
200	[6:00–12:00]	min	681.39	704.11	3.33
200	[6:00–12:00]	avg	1446.11	1553.58	7.43
300	[6:00–8:00]	min	573.19	590.27	2.98
300	[6:00–8:00]	avg	1281.61	1372.04	7.06
300	[6:00–10:00]	min	1122.16	1157.14	3.12
300	[6:00–10:00]	avg	2426.44	2659.18	9.59
300	[6:00–12:00]	min	1370.50	1414.17	3.19
300	[6:00–12:00]	avg	2993.12	3413.19	14.03
340	[6:00–8:00]	min	628.75	645.54	2.67
340	[6:00–8:00]	avg	1438.81	1539.79	7.02
340	[6:00–10:00]	min	1237.02	1270.99	2.75
340	[6:00–10:00]	avg	2734.75	3035.73	11.01
340	[6:00–12:00]	min	1505.53	1549.17	2.90
340	[6:00–12:00]	avg	3383.04	3962.03	17.11
Avg.					5.96

Table 10
Detailed results for the VAR(%) indicator.

Instance			RP	RP^{reb}	VAR(%)
#stations	Time interval	Penalty			
50	[6:00–8:00]	min	46.41	46.65	0.52
50	[6:00–8:00]	avg	95.19	96.69	1.58
50	[6:00–10:00]	min	88.77	89.22	0.52
50	[6:00–10:00]	avg	168.66	172.25	2.13
50	[6:00–12:00]	min	107.11	107.90	0.74
50	[6:00–12:00]	avg	195.00	197.61	1.34
100	[6:00–8:00]	min	111.06	111.42	0.32
100	[6:00–8:00]	avg	236.84	242.49	2.39
100	[6:00–10:00]	min	228.04	228.93	0.39
100	[6:00–10:00]	avg	448.09	455.41	1.63
100	[6:00–12:00]	min	287.46	288.10	0.22
100	[6:00–12:00]	avg	551.81	558.77	1.26
200	[6:00–8:00]	min	265.03	265.62	0.22
200	[6:00–8:00]	avg	603.66	616.50	2.13
200	[6:00–10:00]	min	540.03	541.31	0.24
200	[6:00–10:00]	avg	1148.49	1174.17	2.24
200	[6:00–12:00]	min	681.39	683.63	0.33
200	[6:00–12:00]	avg	1446.11	1474.05	1.93
300	[6:00–8:00]	min	573.19	574.32	0.20
300	[6:00–8:00]	avg	1281.61	1310.18	2.23
300	[6:00–10:00]	min	1122.16	1125.15	0.27
300	[6:00–10:00]	avg	2426.44	2493.60	2.77
300	[6:00–12:00]	min	1370.50	1374.20	0.27
300	[6:00–12:00]	avg	2993.12	3097.43	3.48
340	[6:00–8:00]	min	628.75	629.44	0.11
340	[6:00–8:00]	avg	1438.81	1466.52	1.93
340	[6:00–10:00]	min	1237.02	1239.97	0.24
340	[6:00–10:00]	avg	2734.75	2815.27	2.94
340	[6:00–12:00]	min	1505.53	1508.17	0.18
340	[6:00–12:00]	avg	3383.04	3517.24	3.97
Avg.					1.29

Table 11
Detailed results for the VSS(%) indicator.

Instance			RP	EEV	VSS(%)
#stations	Time interval	Penalty			
50	[6:00–8:00]	min	46.41	85.22	83.64
50	[6:00–8:00]	avg	95.19	558.96	487.22
50	[6:00–10:00]	min	88.77	132.61	49.39
50	[6:00–10:00]	avg	168.66	761.15	351.28
50	[6:00–12:00]	min	107.11	160.39	49.74
50	[6:00–12:00]	avg	195.00	879.17	350.85
100	[6:00–8:00]	min	111.06	159.84	43.92
100	[6:00–8:00]	avg	236.84	1156.66	388.38
100	[6:00–10:00]	min	228.04	320.19	40.41
100	[6:00–10:00]	avg	448.09	2018.89	350.55
100	[6:00–12:00]	min	287.46	390.95	36.00
100	[6:00–12:00]	avg	551.81	2279.22	313.04
200	[6:00–8:00]	min	265.03	333.32	25.77
200	[6:00–8:00]	avg	603.66	2826.94	368.30
200	[6:00–10:00]	min	540.03	684.21	26.70
200	[6:00–10:00]	avg	1148.49	5217.23	354.27
200	[6:00–12:00]	min	681.39	834.45	22.46
200	[6:00–12:00]	avg	1446.11	5758.46	298.20
300	[6:00–8:00]	min	573.19	690.03	20.38
300	[6:00–8:00]	avg	1281.61	5735.50	347.52
300	[6:00–10:00]	min	1122.16	1350.28	20.33
300	[6:00–10:00]	avg	2426.44	9509.12	291.90
300	[6:00–12:00]	min	1370.50	1620.64	18.25
300	[6:00–12:00]	avg	2993.12	10 763.32	259.60
340	[6:00–8:00]	min	628.75	751.92	19.59

(*continued on next page*)

Table 11 (*continued*).

Instance			RP	EEV	VSS(%)
#stations	Time interval	Penalty			
340	[6:00–8:00]	avg	1438.81	6496.04	351.49
340	[6:00–10:00]	min	1237.02	1479.38	19.59
340	[6:00–10:00]	avg	2734.75	10 853.40	296.87
340	[6:00–12:00]	min	1505.53	1764.46	17.20
340	[6:00–12:00]	avg	3383.04	12 249.36	262.08
Avg.					185.50

Table 12
Detailed results for the simulation model run for the SP and the EV solution.

Instance			SP						EV					
#stations	Time interval	Penalty	$\sum_{i\in\mathcal{I}\setminus\{I\}} x_i$	*cong.*(%)	*starv.*(%)	#rebal.	B^+	$\mathbb{E}[FR(\%)]$	$\sum_{i\in\mathcal{I}\setminus\{I\}} x_i$	*cong.*(%)	*starv.*(%)	#rebal.	B^+	$\mathbb{E}[FR(\%)]$
50	[6:00–8:00]	min	27.00	0.00	13.62	0.00	6.00	96.75	4.00	0.00	49.86	0.00	6.43	91.96
50	[6:00–8:00]	avg	65.00	0.00	0.00	0.43	5.14	99.47	4.00	0.00	49.86	0.29	6.14	92.06
50	[6:00–10:00]	min	60.00	0.00	6.06	0.00	13.43	95.64	25.00	0.00	20.55	0.00	14.29	89.49
50	[6:00–10:00]	avg	115.00	0.00	0.32	0.71	12.71	98.88	25.00	0.00	20.55	0.57	13.71	89.67
50	[6:00–12:00]	min	71.00	0.00	9.43	0.00	20.43	94.97	35.00	0.00	26.46	0.00	22.57	90.14
50	[6:00–12:00]	avg	133.00	0.00	1.30	1.29	18.14	98.49	35.00	0.00	26.46	0.86	21.71	90.43
100	[6:00–8:00]	min	65.00	0.00	18.49	0.00	18.29	95.54	21.00	0.00	52.92	0.00	19.57	90.76
100	[6:00–8:00]	avg	164.00	0.00	2.57	0.57	16.71	99.11	21.00	0.00	52.92	0.14	19.43	90.77
100	[6:00–10:00]	min	155.00	0.00	9.63	0.00	46.00	94.08	74.00	0.00	23.35	0.00	47.86	87.31
100	[6:00–10:00]	avg	301.00	0.00	1.22	3.57	42.00	98.79	74.00	0.00	23.35	2.71	45.14	87.73
100	[6:00–12:00]	min	206.00	0.00	9.51	0.00	72.29	93.10	114.00	0.00	20.93	0.00	74.57	86.59
100	[6:00–12:00]	avg	368.00	0.00	1.92	6.00	64.57	97.74	114.00	0.00	20.93	4.14	70.43	87.12
200	[6:00–8:00]	min	165.00	0.00	17.27	0.00	62.00	93.41	75.00	0.00	39.73	0.00	65.14	88.84
200	[6:00–8:00]	avg	399.00	0.00	2.02	2.71	56.14	98.81	75.00	0.00	39.73	1.57	63.71	88.98
200	[6:00–10:00]	min	370.00	0.00	8.28	0.00	147.29	91.26	212.00	0.00	20.56	0.00	151.57	84.93
200	[6:00–10:00]	avg	734.00	0.00	1.26	9.71	136.57	98.03	212.00	0.00	20.56	7.86	145.29	85.44
200	[6:00–12:00]	min	477.00	0.00	10.07	0.00	219.29	89.54	308.00	0.00	19.49	0.00	225.43	84.20
200	[6:00–12:00]	avg	906.00	0.00	2.15	14.14	202.29	96.83	308.00	0.00	19.49	10.00	217.00	84.85
300	[6:00–8:00]	min	365.00	0.00	13.82	0.00	149.29	91.14	208.00	0.00	28.12	0.00	152.86	85.67
300	[6:00–8:00]	avg	834.00	0.00	1.01	8.57	135.43	98.25	208.00	0.00	28.12	6.57	146.29	86.08
300	[6:00–10:00]	min	787.00	0.00	8.25	0.00	363.57	88.30	518.00	0.00	16.27	0.00	366.29	81.55
300	[6:00–10:00]	avg	1483.00	0.06	0.93	18.43	338.29	96.20	520.00	0.00	16.31	13.43	353.29	82.10
300	[6:00–12:00]	min	974.00	0.48	9.14	0.00	525.14	85.94	678.00	0.48	16.78	0.00	532.29	79.92
300	[6:00–12:00]	avg	1785.00	0.88	2.15	25.57	484.57	94.44	680.00	0.48	16.82	20.29	512.00	80.58
340	[6:00–8:00]	min	409.00	0.00	12.71	0.00	159.57	91.82	235.00	0.00	26.23	0.00	163.71	86.43
340	[6:00–8:00]	avg	928.00	0.00	0.91	11.29	143.43	98.50	236.00	0.00	26.23	7.71	156.29	86.87
340	[6:00–10:00]	min	869.00	0.03	7.77	0.00	402.29	88.67	576.00	0.04	16.02	0.00	404.86	82.25
340	[6:00–10:00]	avg	1615.00	0.18	0.94	33.43	362.29	96.55	578.00	0.04	16.07	26.14	381.14	83.25
340	[6:00–12:00]	min	1073.00	0.65	8.62	0.00	573.71	86.32	752.00	0.67	16.41	0.00	581.43	80.90
340	[6:00–12:00]	avg	1942.00	1.24	1.93	37.00	520.86	94.62	754.00	0.67	16.46	28.71	555.14	81.84
Avg.			594.83	0.12	6.11	5.78	177.26	94.71	255.97	0.08	26.25	4.37	184.52	86.29

Table 13
Detailed results for the LUSS (%) indicator.

Instance			SP		ESSV		LUSS	
#stations	Time interval	Penalty	RP	#stations no bikes	ESSV	#stations no bikes	LUSS (%)	Common stations no bikes (%)
50	[6:00–8:00]	min	46.41	27.00	80.82	47.00	74.15	100.00
50	[6:00–8:00]	avg	95.19	11.00	493.04	47.00	417.96	100.00
50	[6:00–10:00]	min	88.77	18.00	99.55	27.00	12.15	100.00
50	[6:00–10:00]	avg	168.66	6.00	356.46	27.00	111.34	100.00
50	[6:00–12:00]	min	107.11	12.00	119.56	24.00	11.62	100.00
50	[6:00–12:00]	avg	195.00	5.00	418.61	24.00	114.67	100.00
100	[6:00–8:00]	min	111.06	49.00	146.58	80.00	31.98	100.00
100	[6:00–8:00]	avg	236.84	16.00	812.90	80.00	243.23	100.00
100	[6:00–10:00]	min	228.04	26.00	246.80	46.00	8.23	100.00
100	[6:00–10:00]	avg	448.09	9.00	872.94	46.00	94.81	100.00
100	[6:00–12:00]	min	287.46	17.00	304.10	34.00	5.79	100.00
100	[6:00–12:00]	avg	551.81	7.00	927.80	33.00	68.14	100.00
200	[6:00–8:00]	min	265.03	87.00	296.29	130.00	11.79	100.00
200	[6:00–8:00]	avg	603.66	26.00	1628.85	129.00	169.83	100.00
200	[6:00–10:00]	min	540.03	52.00	558.06	77.00	3.34	100.00
200	[6:00–10:00]	avg	1148.49	14.00	1923.80	76.00	67.51	100.00
200	[6:00–12:00]	min	681.39	37.00	697.14	56.00	2.31	100.00
200	[6:00–12:00]	avg	1446.11	12.00	2024.76	54.00	40.01	100.00
300	[6:00–8:00]	min	573.19	97.00	602.91	142.00	5.19	100.00
300	[6:00–8:00]	avg	1281.61	26.00	2561.37	135.00	99.85	100.00
300	[6:00–10:00]	min	1122.16	51.00	1142.51	77.00	1.81	100.00
300	[6:00–10:00]	avg	2426.44	15.00	3156.80	71.00	30.10	100.00
300	[6:00–12:00]	min	1370.50	36.00	1396.97	64.00	1.93	100.00
300	[6:00–12:00]	avg	2993.12	13.00	3799.57	59.00	26.94	100.00
340	[6:00–8:00]	min	628.75	121.00	660.52	169.00	5.05	100.00
340	[6:00–8:00]	avg	1438.81	48.00	2795.98	160.00	94.33	100.00

(*continued on next page*)

Table 13 (*continued*).

Instance			SP		ESSV		LUSS	
#stations	Time interval	Penalty	RP	#stations no bikes	ESSV	#stations no bikes	LUSS (%)	Common stations no bikes (%)
340	[6:00–10:00]	min	1237.02	76.00	1254.65	102.00	1.43	100.00
340	[6:00–10:00]	avg	2734.75	37.00	3515.55	96.00	28.55	100.00
340	[6:00–12:00]	min	1505.53	62.00	1531.13	88.00	1.70	100.00
340	[6:00–12:00]	avg	3383.04	35.00	4217.26	82.00	24.66	100.00
Avg.							60.35	

Table 14
Detailed results for the LUDS (%) indicator.

Instance			SP		EIV		Performance	
#stations	Time interval	Penalty	RP	t(s)	EIV	t(s)	LUDS (%)	Δ^t (%)
50	[6:00–8:00]	min	46.41	53.01	46.41	49.78	0.00	−6.09
50	[6:00–8:00]	avg	95.19	49.38	95.19	43.19	0.00	−12.52
50	[6:00–10:00]	min	88.77	45.43	88.77	43.95	0.00	−3.26
50	[6:00–10:00]	avg	168.66	49.48	168.66	42.57	0.00	−13.96
50	[6:00–12:00]	min	107.11	47.70	107.11	51.39	0.00	7.73
50	[6:00–12:00]	avg	195.00	51.19	195.00	47.71	0.00	−6.80
100	[6:00–8:00]	min	111.06	113.22	111.06	101.65	0.00	−10.22
100	[6:00–8:00]	avg	236.84	178.82	236.84	110.27	0.00	−38.34
100	[6:00–10:00]	min	228.04	115.71	228.04	103.59	0.00	−10.48
100	[6:00–10:00]	avg	448.09	109.92	448.09	106.36	0.00	−3.25
100	[6:00–12:00]	min	287.46	115.28	287.82	103.08	0.13	−10.59
100	[6:00–12:00]	avg	551.81	134.99	551.81	107.58	0.00	−20.31
200	[6:00–8:00]	min	265.03	327.60	265.15	223.59	0.05	−31.75
200	[6:00–8:00]	avg	603.66	310.41	603.66	246.88	0.00	−20.47
200	[6:00–10:00]	min	540.03	307.20	540.40	229.71	0.07	−25.22
200	[6:00–10:00]	avg	1148.49	254.63	1148.49	246.08	0.00	−3.36
200	[6:00–12:00]	min	681.39	262.69	681.95	225.40	0.08	−14.20
200	[6:00–12:00]	avg	1446.11	247.55	1446.11	244.45	0.00	−1.25
300	[6:00–8:00]	min	573.19	493.83	574.83	418.24	0.29	−15.31
300	[6:00–8:00]	avg	1281.61	479.42	1281.61	482.28	0.00	0.60
300	[6:00–10:00]	min	1122.16	466.42	1124.81	386.02	0.24	−17.24
300	[6:00–10:00]	avg	2426.44	462.64	2426.44	457.88	0.00	−1.03
300	[6:00–12:00]	min	1370.50	457.35	1373.01	415.96	0.18	−9.05
300	[6:00–12:00]	avg	2993.12	438.91	2993.12	403.98	0.00	−7.96
340	[6:00–8:00]	min	628.75	474.84	630.38	497.96	0.26	4.87
340	[6:00–8:00]	avg	1438.81	501.07	1438.81	538.65	0.00	7.50
340	[6:00–10:00]	min	1237.02	578.62	1239.87	493.54	0.23	−14.70
340	[6:00–10:00]	avg	2734.75	589.23	2734.75	477.50	0.00	−18.96
340	[6:00–12:00]	min	1505.53	568.34	1508.79	493.15	0.22	−13.23
340	[6:00–12:00]	avg	3383.04	584.16	3383.04	467.59	0.00	−19.96
Avg.							0.06	−10.96

Table 15
Detailed results of the sensitivity analysis for the first and second-stage costs obtained by multiplying stockout, excess, and extra inventory penalties by the multiplier u.

Instance				SP	
#stations	Time interval	Penalty	u	First-stage cost	Second-stage cost
50	[6:00–8:00]	min	1	27.00	19.41
50	[6:00–8:00]	min	10	68.00	28.53
50	[6:00–8:00]	min	100	101.00	68.56
50	[6:00–8:00]	min	1000	101.00	216.73
50	[6:00–8:00]	min	10 000	101.00	216.73
50	[6:00–8:00]	avg	1	65.00	30.19
50	[6:00–8:00]	avg	10	100.00	87.09
50	[6:00–8:00]	avg	100	101.00	216.73
50	[6:00–8:00]	avg	1000	101.00	216.73
50	[6:00–8:00]	avg	10 000	101.00	216.73
50	[6:00–10:00]	min	1	60.00	28.77
50	[6:00–10:00]	min	10	117.00	50.78
50	[6:00–10:00]	min	100	161.00	157.34

(*continued on next page*)

Table 15 (*continued*).

Instance				SP	
#stations	Time interval	Penalty	*u*	First-stage cost	Second-stage cost
50	[6:00–10:00]	min	1000	169.00	532.05
50	[6:00–10:00]	min	10 000	169.00	540.68
50	[6:00–10:00]	avg	1	115.00	53.66
50	[6:00–10:00]	avg	10	164.00	203.49
50	[6:00–10:00]	avg	100	169.00	532.67
50	[6:00–10:00]	avg	1000	169.00	547.33
50	[6:00–10:00]	avg	10 000	169.00	693.93
50	[6:00–12:00]	min	1	71.00	36.11
50	[6:00–12:00]	min	10	135.00	60.71
50	[6:00–12:00]	min	100	177.00	200.01
50	[6:00–12:00]	min	1000	183.00	664.26
50	[6:00–12:00]	min	10 000	183.00	762.00
50	[6:00–12:00]	avg	1	133.00	62.00
50	[6:00–12:00]	avg	10	177.00	258.20
50	[6:00–12:00]	avg	100	183.00	671.54
50	[6:00–12:00]	avg	1000	183.00	839.23
50	[6:00–12:00]	avg	10 000	183.00	2516.14
100	[6:00–8:00]	min	1	65.00	46.06
100	[6:00–8:00]	min	10	160.00	63.70
100	[6:00–8:00]	min	100	237.00	180.46
100	[6:00–8:00]	min	1000	245.00	538.47
100	[6:00–8:00]	min	10 000	245.00	551.50
100	[6:00–8:00]	avg	1	164.00	72.84
100	[6:00–8:00]	avg	10	240.00	240.55
100	[6:00–8:00]	avg	100	245.00	541.49
100	[6:00–8:00]	avg	1000	245.00	563.73
100	[6:00–8:00]	avg	10 000	245.00	764.68
100	[6:00–10:00]	min	1	155.00	73.04
100	[6:00–10:00]	min	10	292.00	123.61
100	[6:00–10:00]	min	100	408.00	416.20
100	[6:00–10:00]	min	1000	412.00	1459.09
100	[6:00–10:00]	min	10 000	412.00	4551.82
100	[6:00–10:00]	avg	1	301.00	147.09
100	[6:00–10:00]	avg	10	409.00	594.11
100	[6:00–10:00]	avg	100	412.00	1828.72
100	[6:00–10:00]	avg	1000	412.00	8720.24
100	[6:00–10:00]	avg	10 000	412.00	77 274.48
100	[6:00–12:00]	min	1	206.00	81.46
100	[6:00–12:00]	min	10	355.00	161.40
100	[6:00–12:00]	min	100	478.00	565.97
100	[6:00–12:00]	min	1000	491.00	2030.77
100	[6:00–12:00]	min	10 000	491.00	8703.01
100	[6:00–12:00]	avg	1	368.00	183.81
100	[6:00–12:00]	avg	10	487.00	781.20
100	[6:00–12:00]	avg	100	491.00	2781.40
100	[6:00–12:00]	avg	1000	491.00	16 989.22
100	[6:00–12:00]	avg	10 000	491.00	158 570.06
200	[6:00–8:00]	min	1	165.00	100.03
200	[6:00–8:00]	min	10	351.00	159.04
200	[6:00–8:00]	min	100	551.00	485.25
200	[6:00–8:00]	min	1000	586.00	1899.36
200	[6:00–8:00]	min	10 000	586.00	5451.72
200	[6:00–8:00]	avg	1	399.00	204.66
200	[6:00–8:00]	avg	10	577.00	756.61
200	[6:00–8:00]	avg	100	586.00	3493.86
200	[6:00–8:00]	avg	1000	586.00	11 244.10
200	[6:00–8:00]	avg	10 000	586.00	88 028.94

(*continued on next page*)

Table 15 (*continued*).

Instance				SP	
#stations	Time interval	Penalty	u	First-stage cost	Second-stage cost
200	[6:00–10:00]	min	1	370.00	170.03
200	[6:00–10:00]	min	10	668.00	293.38
200	[6:00–10:00]	min	100	940.00	1138.23
200	[6:00–10:00]	min	1000	997.00	5175.90
200	[6:00–10:00]	min	10 000	997.00	32 940.12
200	[6:00–10:00]	avg	1	734.00	414.49
200	[6:00–10:00]	avg	10	980.00	1901.21
200	[6:00–10:00]	avg	100	999.00	10 818.53
200	[6:00–10:00]	avg	1000	999.00	86 309.35
200	[6:00–10:00]	avg	10 000	999.00	840 691.67
200	[6:00–12:00]	min	1	477.00	204.39
200	[6:00–12:00]	min	10	829.00	377.31
200	[6:00–12:00]	min	100	1176.00	1483.66
200	[6:00–12:00]	min	1000	1245.00	6614.52
200	[6:00–12:00]	min	10 000	1245.00	44 857.33
200	[6:00–12:00]	avg	1	906.00	540.11
200	[6:00–12:00]	avg	10	1227.00	2422.51
200	[6:00–12:00]	avg	100	1245.00	13 519.62
200	[6:00–12:00]	avg	1000	1245.00	112 470.89
200	[6:00–12:00]	avg	10 000	1245.00	1101457.09
300	[6:00–8:00]	min	1	365.00	208.19
300	[6:00–8:00]	min	10	744.00	308.72
300	[6:00–8:00]	min	100	1109.00	1033.81
300	[6:00–8:00]	min	1000	1179.00	5028.29
300	[6:00–8:00]	min	10 000	1179.00	27 022.08
300	[6:00–8:00]	avg	1	834.00	447.61
300	[6:00–8:00]	avg	10	1158.00	1819.65
300	[6:00–8:00]	avg	100	1179.00	10 923.80
300	[6:00–8:00]	avg	1000	1179.00	68 723.49
300	[6:00–8:00]	avg	10 000	1179.00	627 786.41
300	[6:00–10:00]	min	1	787.00	335.16
300	[6:00–10:00]	min	10	1350.00	593.23
300	[6:00–10:00]	min	100	1857.00	2602.24
300	[6:00–10:00]	min	1000	1972.00	14 631.73
300	[6:00–10:00]	min	10 000	1972.00	112 549.99
300	[6:00–10:00]	avg	1	1483.00	943.44
300	[6:00–10:00]	avg	10	1920.00	4893.95
300	[6:00–10:00]	avg	100	1972.00	34 557.50
300	[6:00–10:00]	avg	1000	1972.00	300 466.54
300	[6:00–10:00]	avg	10 000	1972.00	2955130.32
300	[6:00–12:00]	min	1	974.00	396.50
300	[6:00–12:00]	min	10	1612.00	778.04
300	[6:00–12:00]	min	100	2240.00	3416.46
300	[6:00–12:00]	min	1000	2345.00	19 662.00
300	[6:00–12:00]	min	10 000	2346.00	161 695.33
300	[6:00–12:00]	avg	1	1785.00	1208.12
300	[6:00–12:00]	avg	10	2305.00	6360.33
300	[6:00–12:00]	avg	100	2348.00	44 937.40
300	[6:00–12:00]	avg	1000	2349.00	405 716.23
300	[6:00–12:00]	avg	10 000	2349.00	4010660.67
340	[6:00–8:00]	min	1	409.00	219.75
340	[6:00–8:00]	min	10	812.00	340.41
340	[6:00–8:00]	min	100	1221.00	1185.77
340	[6:00–8:00]	min	1000	1305.00	4946.47
340	[6:00–8:00]	min	10 000	1305.00	23 629.30
340	[6:00–8:00]	avg	1	928.00	510.81
340	[6:00–8:00]	avg	10	1287.00	2212.99
340	[6:00–8:00]	avg	100	1305.00	10 679.84
340	[6:00–8:00]	avg	1000	1305.00	66 327.93
340	[6:00–8:00]	avg	10 000	1305.00	605 616.25

(*continued on next page*)

Table 15 (*continued*).

Instance				SP	
#stations	Time interval	Penalty	u	First-stage cost	Second-stage cost
340	[6:00–10:00]	min	1	869.00	368.02
340	[6:00–10:00]	min	10	1471.00	655.71
340	[6:00–10:00]	min	100	2002.00	2979.09
340	[6:00–10:00]	min	1000	2136.00	15 851.01
340	[6:00–10:00]	min	10 000	2137.00	126 251.53
340	[6:00–10:00]	avg	1	1615.00	1119.75
340	[6:00–10:00]	avg	10	2101.00	6005.97
340	[6:00–10:00]	avg	100	2139.00	40 711.70
340	[6:00–10:00]	avg	1000	2139.00	372 184.98
340	[6:00–10:00]	avg	10 000	2139.00	3684400.69
340	[6:00–12:00]	min	1	1073.00	432.53
340	[6:00–12:00]	min	10	1758.00	856.32
340	[6:00–12:00]	min	100	2425.00	3948.87
340	[6:00–12:00]	min	1000	2523.00	22 597.97
340	[6:00–12:00]	min	10 000	2525.00	190 381.94
340	[6:00–12:00]	avg	1	1942.00	1441.04
340	[6:00–12:00]	avg	10	2502.00	7856.46
340	[6:00–12:00]	avg	100	2526.00	55 226.60
340	[6:00–12:00]	avg	1000	2526.00	515 360.58
340	[6:00–12:00]	avg	10 000	2526.00	5115164.44

Table 16
Detailed results of the sensitivity analysis for the first and second-stage costs obtained by multiplying rebalancing, stockout, excess, and extra inventory penalties by the multiplier u.

Instance				SP	
#stations	Time interval	Penalty	u	First-stage cost	Second-stage cost
50	[6:00–8:00]	min	1	27.00	19.41
50	[6:00–8:00]	min	10	74.00	27.07
50	[6:00–8:00]	min	100	101.00	140.82
50	[6:00–8:00]	min	1000	101.00	1408.22
50	[6:00–8:00]	min	10 000	101.00	14 082.18
50	[6:00–8:00]	avg	1	65.00	30.19
50	[6:00–8:00]	avg	10	100.00	129.80
50	[6:00–8:00]	avg	100	101.00	1288.63
50	[6:00–8:00]	avg	1000	101.00	12 886.26
50	[6:00–8:00]	avg	10 000	101.00	128 862.57
50	[6:00–10:00]	min	1	60.00	28.77
50	[6:00–10:00]	min	10	122.00	53.45
50	[6:00–10:00]	min	100	164.00	340.75
50	[6:00–10:00]	min	1000	169.00	3375.57
50	[6:00–10:00]	min	10 000	169.00	33 755.68
50	[6:00–10:00]	avg	1	115.00	53.66
50	[6:00–10:00]	avg	10	166.00	312.62
50	[6:00–10:00]	avg	100	169.00	3097.74
50	[6:00–10:00]	avg	1000	169.00	30 977.40
50	[6:00–10:00]	avg	10 000	169.00	309 773.95
50	[6:00–12:00]	min	1	71.00	36.11
50	[6:00–12:00]	min	10	140.00	66.20
50	[6:00–12:00]	min	100	179.00	475.44
50	[6:00–12:00]	min	1000	187.00	4709.83
50	[6:00–12:00]	min	10 000	187.00	47 098.30
50	[6:00–12:00]	avg	1	133.00	62.00
50	[6:00–12:00]	avg	10	179.00	414.44
50	[6:00–12:00]	avg	100	183.00	4106.95
50	[6:00–12:00]	avg	1000	184.00	41 065.51
50	[6:00–12:00]	avg	10 000	186.00	410 635.13
100	[6:00–8:00]	min	1	65.00	46.06
100	[6:00–8:00]	min	10	175.00	57.61
100	[6:00–8:00]	min	100	240.00	316.18
100	[6:00–8:00]	min	1000	245.00	3119.81
100	[6:00–8:00]	min	10 000	245.00	31 198.13

(*continued on next page*)

Table 16 (*continued*).

Instance				SP	
#stations	Time interval	Penalty	u	First-stage cost	Second-stage cost
100	[6:00–8:00]	avg	1	164.00	72.84
100	[6:00–8:00]	avg	10	241.00	385.96
100	[6:00–8:00]	avg	100	245.00	3822.96
100	[6:00–8:00]	avg	1000	245.00	38 229.64
100	[6:00–8:00]	avg	10 000	245.00	382 296.38
100	[6:00–10:00]	min	1	155.00	73.04
100	[6:00–10:00]	min	10	304.00	129.77
100	[6:00–10:00]	min	100	414.00	786.62
100	[6:00–10:00]	min	1000	424.00	7796.41
100	[6:00–10:00]	min	10 000	424.00	77 964.10
100	[6:00–10:00]	avg	1	301.00	147.09
100	[6:00–10:00]	avg	10	410.00	979.54
100	[6:00–10:00]	avg	100	416.00	9769.17
100	[6:00–10:00]	avg	1000	424.00	97 635.72
100	[6:00–10:00]	avg	10 000	424.00	976 357.25
100	[6:00–12:00]	min	1	206.00	81.46
100	[6:00–12:00]	min	10	374.00	165.96
100	[6:00–12:00]	min	100	498.00	1136.39
100	[6:00–12:00]	min	1000	518.00	11 241.57
100	[6:00–12:00]	min	10 000	518.00	112 415.69
100	[6:00–12:00]	avg	1	368.00	183.81
100	[6:00–12:00]	avg	10	487.00	1333.85
100	[6:00–12:00]	avg	100	506.00	13 273.50
100	[6:00–12:00]	avg	1000	518.00	132 674.09
100	[6:00–12:00]	avg	10 000	518.00	1326740.92
200	[6:00–8:00]	min	1	165.00	100.03
200	[6:00–8:00]	min	10	383.00	155.63
200	[6:00–8:00]	min	100	568.00	713.09
200	[6:00–8:00]	min	1000	591.00	6973.97
200	[6:00–8:00]	min	10 000	591.00	69 739.70
200	[6:00–8:00]	avg	1	399.00	204.66
200	[6:00–8:00]	avg	10	579.00	1206.27
200	[6:00–8:00]	avg	100	588.00	11 994.34
200	[6:00–8:00]	avg	1000	590.00	119 935.37
200	[6:00–8:00]	avg	10 000	591.00	1199350.29
200	[6:00–10:00]	min	1	370.00	170.03
200	[6:00–10:00]	min	10	718.00	290.56
200	[6:00–10:00]	min	100	980.00	1733.44
200	[6:00–10:00]	min	1000	1028.00	17 027.47
200	[6:00–10:00]	min	10 000	1028.00	170 274.66
200	[6:00–10:00]	avg	1	734.00	414.49
200	[6:00–10:00]	avg	10	983.00	3008.78
200	[6:00–10:00]	avg	100	1008.00	29 924.23
200	[6:00–10:00]	avg	1000	1027.00	299 136.81
200	[6:00–10:00]	avg	10 000	1028.00	2991365.32
200	[6:00–12:00]	min	1	477.00	204.39
200	[6:00–12:00]	min	10	889.00	384.82
200	[6:00–12:00]	min	100	1254.00	2362.29
200	[6:00–12:00]	min	1000	1311.00	23 242.26
200	[6:00–12:00]	min	10 000	1311.00	232 422.64
200	[6:00–12:00]	avg	1	906.00	540.11
200	[6:00–12:00]	avg	10	1235.00	3976.79
200	[6:00–12:00]	avg	100	1279.00	39 618.78
200	[6:00–12:00]	avg	1000	1309.00	396 012.33
200	[6:00–12:00]	avg	10 000	1311.00	3960112.30
300	[6:00–8:00]	min	1	365.00	208.19
300	[6:00–8:00]	min	10	803.00	311.25
300	[6:00–8:00]	min	100	1146.00	1580.16
300	[6:00–8:00]	min	1000	1191.00	15 527.05
300	[6:00–8:00]	min	10 000	1191.00	155 270.49

(*continued on next page*)

Table 16 (*continued*).

Instance				SP	
#stations	Time interval	Penalty	u	First-stage cost	Second-stage cost
300	[6:00–8:00]	avg	1	834.00	447.61
300	[6:00–8:00]	avg	10	1163.00	2939.18
300	[6:00–8:00]	avg	100	1183.00	29 240.24
300	[6:00–8:00]	avg	1000	1193.00	292 337.69
300	[6:00–8:00]	avg	10 000	1193.00	2923376.87
300	[6:00–10:00]	min	1	787.00	335.16
300	[6:00–10:00]	min	10	1460.00	617.80
300	[6:00–10:00]	min	100	1980.00	3923.73
300	[6:00–10:00]	min	1000	2068.00	38 670.13
300	[6:00–10:00]	min	10 000	2069.00	386 697.95
300	[6:00–10:00]	avg	1	1483.00	943.44
300	[6:00–10:00]	avg	10	1936.00	7319.28
300	[6:00–10:00]	avg	100	2023.00	72 758.28
300	[6:00–10:00]	avg	1000	2068.00	727 343.21
300	[6:00–10:00]	avg	10 000	2070.00	7273417.49
300	[6:00–12:00]	min	1	974.00	396.50
300	[6:00–12:00]	min	10	1769.00	832.82
300	[6:00–12:00]	min	100	2360.00	5688.90
300	[6:00–12:00]	min	1000	2475.00	56 189.05
300	[6:00–12:00]	min	10 000	2485.00	561 819.48
300	[6:00–12:00]	avg	1	1785.00	1208.12
300	[6:00–12:00]	avg	10	2317.00	9626.41
300	[6:00–12:00]	avg	100	2427.00	95 842.06
300	[6:00–12:00]	avg	1000	2493.00	958 059.05
300	[6:00–12:00]	avg	10 000	2499.00	9580545.00
340	[6:00–8:00]	min	1	409.00	219.75
340	[6:00–8:00]	min	10	876.00	343.10
340	[6:00–8:00]	min	100	1274.00	1727.73
340	[6:00–8:00]	min	1000	1319.00	16 999.51
340	[6:00–8:00]	min	10 000	1320.00	169 993.86
340	[6:00–8:00]	avg	1	928.00	510.81
340	[6:00–8:00]	avg	10	1293.00	3495.46
340	[6:00–8:00]	avg	100	1308.00	34 838.73
340	[6:00–8:00]	avg	1000	1320.00	348 318.27
340	[6:00–8:00]	avg	10 000	1320.00	3483182.69
340	[6:00–10:00]	min	1	869.00	368.02
340	[6:00–10:00]	min	10	1603.00	676.05
340	[6:00–10:00]	min	100	2128.00	4527.41
340	[6:00–10:00]	min	1000	2241.00	44 554.16
340	[6:00–10:00]	min	10 000	2247.00	445 512.84
340	[6:00–10:00]	avg	1	1615.00	1119.75
340	[6:00–10:00]	avg	10	2111.00	8925.33
340	[6:00–10:00]	avg	100	2182.00	88 890.92
340	[6:00–10:00]	avg	1000	2253.00	888 535.42
340	[6:00–10:00]	avg	10 000	2255.00	8885348.83
340	[6:00–12:00]	min	1	1073.00	432.53
340	[6:00–12:00]	min	10	1934.00	936.65
340	[6:00–12:00]	min	100	2563.00	6720.75
340	[6:00–12:00]	min	1000	2692.00	66 407.45
340	[6:00–12:00]	min	10 000	2694.00	664 063.28
340	[6:00–12:00]	avg	1	1942.00	1441.04
340	[6:00–12:00]	avg	10	2507.00	11 802.96
340	[6:00–12:00]	avg	100	2604.00	117 625.97
340	[6:00–12:00]	avg	1000	2697.00	1175739.32
340	[6:00–12:00]	avg	10 000	2699.00	11757385.85

References

Albiński, Szymon, Fontaine, Pirmin, Minner, Stefan, 2018. Performance analysis of a hybrid bike sharing system: A service-level-based approach under censored demand observations. Transp. Res. E 116, 59–69.

Alvarez-Valdes, Ramon, Belenguer, Jose M., Benavent, Enrique, Bermudez, Jose D., Muñoz, Facundo, Vercher, Enriqueta, Verdejo, Francisco, 2016. Optimizing the level of service quality of a bike-sharing system. Omega 62, 163–175.

Benchimol, Mike, Benchimol, Pascal, Chappert, Benoît, De La Taille, Arnaud, Laroche, Fabien, Meunier, Frédéric, Robinet, Ludovic, 2011. Balancing the stations of a self service "bike hire" system. RAIRO-Oper. Res. 45 (1), 37–61.

Berbeglia, Gerardo, Cordeau, Jean-François, Gribkovskaia, Irina, Laporte, Gilbert, 2007. Static pickup and delivery problems: a classification scheme and survey. Top 15 (1), 1–31.

Bertazzi, Luca, Maggioni, Francesca, 2018. A stochastic multi-stage fixed charge transportation problem: Worst-case analysis of the rolling horizon approach. European J. Oper. Res. 267 (2), 555–569.

Birge, John R., Louveaux, François, 2011. Introduction to Stochastic Programming. Springer Science & Business Media.
Brinkmann, Jan, Ulmer, Marlin W., Mattfeld, Dirk C., 2015. Short-term strategies for stochastic inventory routing in bike sharing systems. Transp. Res. Procedia 10, 364–373.
Brinkmann, Jan, Ulmer, Marlin W., Mattfeld, Dirk C., 2016. Inventory routing for bike sharing systems. Transp. Res. Procedia 19, 316–327.
Brinkmann, Jan, Ulmer, Marlin W., Mattfeld, Dirk C., 2019. Dynamic lookahead policies for stochastic-dynamic inventory routing in bike sharing systems. Comput. Oper. Res. 106, 260–279.
Brinkmann, Jan, Ulmer, Marlin W., Mattfeld, Dirk C., 2020. The multi-vehicle stochastic-dynamic inventory routing problem for bike sharing systems. Bus. Res. 13 (1), 69–92.
Bruck, Bruno P., Cruz, Fábio, Iori, Manuel, Subramanian, Anand, 2019. The static bike sharing rebalancing problem with forbidden temporary operations. Transp. Sci. 53 (3), 882–896.
Bulhões, Teobaldo, Subramanian, Anand, Erdoğan, Güneş, Laporte, Gilbert, 2018. The static bike relocation problem with multiple vehicles and visits. European J. Oper. Res. 264 (2), 508–523.
Chemla, Daniel, Meunier, Frédéric, Calvo, Roberto Wolfler, 2013. Bike sharing systems: Solving the static rebalancing problem. Discrete Optim. 10 (2), 120–146.
Contardo, Claudio, Morency, Catherine, Rousseau, Louis-Martin, 2012. Balancing a Dynamic Public Bike-sharing System, vol. 4, Cirrelt Montreal, Canada.
Cruz, Fábio, Subramanian, Anand, Bruck, Bruno P., Iori, Manuel, 2017. A heuristic algorithm for a single vehicle static bike sharing rebalancing problem. Comput. Oper. Res. 79, 19–33.
Datner, Sharon, Raviv, Tal, Tzur, Michal, Chemla, Daniel, 2017. Setting inventory levels in a bike sharing network. Transp. Sci..
de Chardon, Cyrille Médard, Caruso, Geoffrey, Thomas, Isabelle, 2016. Bike-share rebalancing strategies, patterns, and purpose. J. Transp. Geogr. 55, 22–39.
Dell'Amico, Mauro, Hadjicostantinou, Eleni, Iori, Manuel, Novellani, Stefano, 2014. The bike sharing rebalancing problem: Mathematical formulations and benchmark instances. Omega 45, 7–19.
Dell'Amico, Mauro, Iori, Manuel, Novellani, Stefano, Stützle, Thomas, 2016. A destroy and repair algorithm for the Bike sharing Rebalancing Problem. Comput. Oper. Res. 71, 149–162.
Dell'Amico, Mauro, Iori, Manuel, Novellani, Stefano, Subramanian, Anand, 2018. The bike sharing rebalancing problem with stochastic demands. Transp. Res. B 118, 362–380.
Divvy Bikes, 2024. https://divvybikes.com/valet.
Erdoğan, Güneş, Battarra, Maria, Calvo, Roberto Wolfler, 2015. An exact algorithm for the static rebalancing problem arising in bicycle sharing systems. European J. Oper. Res. 245 (3), 667–679.
Erdoğan, Güneş, Laporte, Gilbert, Calvo, Roberto Wolfler, 2014. The static bicycle relocation problem with demand intervals. European J. Oper. Res. 238 (2), 451–457.
Erera, Alan L., Savelsbergh, Martin, Uyar, Emrah, 2009. Fixed routes with backup vehicles for stochastic vehicle routing problems with time constraints. Networks 54 (4), 270–283.
Forma, Iris A., Raviv, Tal, Tzur, Michal, 2015. A 3-step math heuristic for the static repositioning problem in bike-sharing systems. Transp. Res. B 71, 230–247.
Fu, Chenyi, Zhu, Ning, Ma, Shoufeng, Liu, Ronghui, 2022. A two-stage robust approach to integrated station location and rebalancing vehicle service design in bike-sharing systems. European J. Oper. Res. 298 (3), 915–938.
Gammelli, Daniele, Peled, Inon, Rodrigues, Filipe, Pacino, Dario, Kurtaran, Haci A., Pereira, Francisco C., 2020. Estimating latent demand of shared mobility through censored Gaussian processes. Transp. Res. C 120, 102775.
Ghosh, Supriyo, Varakantham, Pradeep, Adulyasak, Yossiri, Jaillet, Patrick, 2017. Dynamic repositioning to reduce lost demand in bike sharing systems. J. Artificial Intelligence Res. 58, 387–430.
Ho, Sin C., Szeto, W.Y., 2014. Solving a static repositioning problem in bike-sharing systems using iterated tabu search. Transp. Res. E 69, 180–198.
Kall, Peter, Wallace, Stein W., 1994. Stochastic Programming. Springer.
Kaut, Michal, Wallace, Stein W., 2007. Evaluation of scenario-generation methods for stochastic programming. Pac. J. Optim. 3 (2), 257–271.
King, Alan J., Wallace, Stein W., 2012. Modeling with Stochastic Programming. Springer Science & Business Media.
Legros, Benjamin, 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, Xiang, Wang, Xianzhe, Feng, Ziyan, 2024. Dynamic repositioning in bike-sharing systems with uncertain demand: An improved rolling horizon framework. Omega.
Lu, Chung-Cheng, 2016. Robust multi-period fleet allocation models for bike-sharing systems. Netw. Spat. Econ. 16 (1), 61–82.
Lv, Chang, Zhang, Chaoyong, Lian, Kunlei, Ren, Yaping, Meng, Leilei, 2020. A hybrid algorithm for the static bike-sharing re-positioning problem based on an effective clustering strategy. Transp. Res. B 140, 1–21.
Maggioni, Francesca, Cagnolari, Matteo, Bertazzi, Luca, Wallace, Stein W., 2019. Stochastic optimization models for a bike-sharing problem with transshipment. European J. Oper. Res. 276 (1), 272–283.
Maggioni, Francesca, Wallace, Stein W., 2012. Analyzing the quality of the expected value solution in stochastic programming. Ann. Oper. Res. 200 (1), 37–54.
Nair, Rahul, Miller-Hooks, Elise, 2011. Fleet management for vehicle sharing operations. Transp. Sci. 45 (4), 524–540.
Negahban, Ashkan, 2019. Simulation-based estimation of the real demand in bike-sharing systems in the presence of censoring. European J. Oper. Res. 277 (1), 317–332.
Neumann-Saavedra, Bruno A., Vogel, Patrick, Mattfeld, Dirk C., 2015. Anticipatory service network design of bike sharing systems. Transp. Res. Procedia 10, 355–363.
PBSC, 2021. https://www.pbsc.com/blog/2021/10/the-meddin-bike-sharing-world-map.
Raviv, Tal, Kolka, Ofer, 2013. Optimal inventory management of a bike-sharing station. IIE Trans. 45 (10), 1077–1093.
Raviv, Tal, Tzur, Michal, Forma, Iris A., 2013. Static repositioning in a bike-sharing system: models and solution approaches. EURO J. Transp. Logist. 2 (3), 187–229.
Regue, Robert, Recker, Will, 2014. Proactive vehicle routing with inferred demand to solve the bikesharing rebalancing problem. Transp. Res. E 72, 192–209.
Ren, Yaping, Meng, Leilei, Zhao, Fu, Zhang, Chaoyong, Guo, Hongfei, Tian, Ying, Tong, Wen, Sutherland, John W., 2020. An improved general variable neighborhood search for a static bike-sharing rebalancing problem considering the depot inventory. Expert Syst. Appl. 160, 113752.
Schuijbroek, Jasper, Hampshire, Robert C., Van Hoeve, W.-J., 2017. Inventory rebalancing and vehicle routing in bike sharing systems. European J. Oper. Res. 257 (3), 992–1004.
S.F. Gov, 2024. https://www.sf.gov/news/san-francisco-announces-expansion-lyfts-bike-share-program-golden-gate-park.
Shui, C.S., Szeto, W.Y., 2020. A review of bicycle-sharing service planning problems. Transp. Res. C 117, 102648.
Vogel, Patrick, 2016. Service network design of bike sharing systems. In: Service Network Design of Bike Sharing Systems. Springer.
Vogel, Patrick, Saavedra, Bruno A. Neumann, Mattfeld, Dirk C., 2014. A hybrid metaheuristic to solve the resource allocation problem in bike sharing systems. In: International Workshop on Hybrid Metaheuristics. Springer.