A two-stage stochastic optimization model for the bike-sharing allocation and rebalancing problem

Rossana Cavagnini[1], Francesca Maggioni[2], Luca Bertazzi[3], and Mike Hewitt[4]

[1]Deutsche Post Chair – Optimization of Distribution Networks, RWTH Aachen University, Germany

[2]Department of Management, Information and Production Engineering, University of Bergamo, Italy

[3]Department of Economics and Management, University of Brescia, Italy

[4]Department of Information Systems and Operations Management, Quinlan School of Business, Loyola University Chicago, USA

cavagnini@dpo.rwth-aachen.de, francesca.maggioni@unibg.it, luca.bertazzi@unibs.it, mhewitt3@luc.edu

September 7, 2023

Abstract

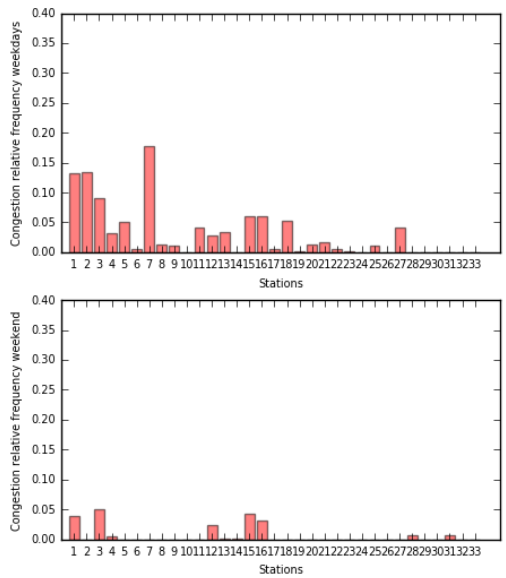
The *Bike-sharing allocation and rebalancing problem* is the problem of determining the initial daily allocation of bikes to stations in a bike-sharing system composed of one depot and multiple capacitated stations, in which bikes can be rebalanced at a point in time during the day. Due to the uncertain demand in each station, we propose a two-stage stochastic programming formulation, where the allocation is made at the first stage and the recourse decisions related to rebalancing are made at the second stage. The impact of the stochastic demand on the problem solution is examined, showing the benefits of the proposed methodology with respect to the deterministic formulation. A procedure for solving the stochastic program using information from the deterministic solution is derived showing a significant reduction in solution time without losing solution quality. The proposed approach is finally benchmarked on the real bike-sharing system of the city of San Francisco.

Keywords: Bike sharing, Rebalancing, Stochastic programming, Stochastic solution analysis.

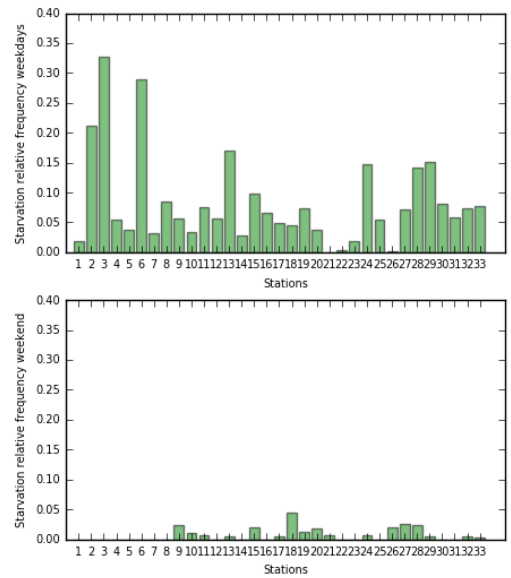
1 Introduction

Bike-sharing systems are becoming more prevalent and 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 for use (typically via a rental agreement) by different individuals throughout the day. These systems typically consist of a depot (or set of depots), wherein bikes are stored at the beginning of the day, and multiple stations located throughout the city, from which an individual can withdraw a bike for a (usually short) journey and then return that bike to a station that may be different from the station from which it was withdrawn. These stations typically have a fixed number of slots for holding bikes, although their capacity can be expanded on a temporary basis. Finally, these stations often have technology that enables them to communicate information regarding their status (e.g., how many bikes are currently there) to a central manager/planner. Typically, 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 can include determining the number, location, and capacity of stations for bike rental and return, whereas tactical decisions can include 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 (the so-called "valet service"). The service provider has to decide first how to allocate bikes to stations and then how to re-distribute bikes among stations, to rebalance the system. Rebalancing is performed at specific time periods during the day, through a capacitated vehicle that travels a given route. The service provider tries to limit the cases in which an individual arrives at a station in hopes of renting a bike, but none is available (a situation we refer to as "*starvation*"). On the other hand, the service provider tries to limit the cases in which an individual seeks to return a bike to a station, but the station is already full (a situation we refer to as "*congestion*"). Both cases negatively impact the user's experience with the bike-sharing service, as they both (potentially) require the user to travel to another station. At the same time, they are somewhat competing objectives, as the more bikes allocated to a station, the less the likelihood of a starvation event, but the greater the likelihood of a congestion event. To understand the magnitude of these phenomena, Figure 1 shows the percentage of minutes with congestion and starvation on weekdays and weekends, by station, registered in the time interval between 6 a.m. and 11:59 a.m. for the months of May through August, 2016 for the San Francisco bike-sharing system. Figures 1a and 1b show that congestion and starvation are more likely during the week, with some stations seeing a congestion and starvation event 20% and 30% of the time, respectively.



(a) Congestion frequency during weekdays and weekend



(b) Starvation frequency during weekdays and weekend

Figure 1: Congestion and starvation average frequencies in the San Francisco bike-sharing system for year 2016, considering the months from May to August and the time interval between 6:00 a.m. and 11:59 a.m.

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 damages and deterioration), as well as the number of bikes that are redistributed through the day.

We formulate a two-stage stochastic optimization model to represent this problem. The purpose of this model is determining an effective initial allocation of bikes to stations and approximating the possibility of rebalancing bikes with a static second-stage problem. We show that explicitly modeling uncertainty in demand leads to improvements along multiple performance dimensions over using a deterministic model. Specifically, our stochastic program includes objectives that measure each of the performance dimensions mentioned above (congestion, starvation, fleet size, rebalancing frequency). The impact of the stochastic demand on problem solution is examined, showing the benefits of the proposed methodology on solution quality when compared to solving the deterministic formulation. Nevertheless, by assessing the upgradeability of solutions from a deterministic version of this problem (see Maggioni and Wallace [2012]), we derive a heuristic procedure for solving the stochastic program that significantly reduces its solution time without losing solution quality.

To assess the performance of the initial allocation plan produced by the stochastic program, we compare it with the one resulting by considering the number of available bikes at stations in the morning in the real San Francisco bike-sharing system. The two plans are compared across multiple dimensions (frequency of congestion and starvation, and total number of bikes allocated and rebalanced between stations) in the context of a simulation of one week of operation for the San Francisco bike-sharing system. The system is integrated with a mobile app, which allows the users to know in real time the number of available slots in all bike stations throughout the city. Numerical results show that the solution to the stochastic program outperforms the real San Francisco implemented allocation plan on all dimensions.

We carry out computational experiments to answer the following questions:

1. How to generate realistic demand scenarios?

2. What is the minimum number of scenarios to be considered to describe the stochastic demand?

3. What is the Value of Stochastic Solution (VSS)?

4. What is the quality of the Expected Value (EV) solution when solving the stochastic model?

5. How does the actual number of bikes available in the morning in the San Francisco

bike-sharing system perform compared to the ones obtained by optimally solving the stochastic and deterministic models?

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 model is formulated in Section 4. In Section 5, we present computational results and relevant analyses that allow us to give an answer to the previous questions. Finally, Section 6 provides conclusions and suggestions for future works.

2 Literature review

In this section, we review the papers studying station-based bike-sharing systems. Specifically, we review the papers focusing exclusively on the problem of determining the initial inventory level of bikes at stations in Section 2.1, the papers studying exclusively the problem of rebalancing in Section 2.2, and, the papers considering both problems jointly in Section 2.3. Table 1 summarizes the surveyed papers. In the following, we refer to “initial inventory”as the amount of bikes to be allocated at stations initially and to “target inventory”as the amount of bikes that should be at stations after rebalancing. 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].

2.1 Initial station inventory level problems

Raviv and Kolka [2013] focus on the problem of determining the optimal initial inventory level at a single bike-sharing station. They minimize a user dissatisfaction measure, expressed as a function of the bikes and lockers shortage events. The proposed model omits interdependencies among stations, both with reference to the demand and arrival process, 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 initial 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 one considering interactions between stations in setting the initial inventory levels and, to the best of our knowledge, it is the only one considering a stochastic demand process, it presents three differences with respect to our work: (1) 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), (2) it does not consider the possibility that rebalancing may take place, and (3) 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.

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 *PDP* (Pick up and Delivery Problem) with transshipment, since each rebalancing flow has exactly one pickup station and one delivery station, and these stations are determined by the order according to which they are visited.

However, this variant of the PDP does not consider the impact on the customer service level (e.g., dissatisfaction for congestion and starvation events). The same limitation can be found in the following papers which also use pre-determined target levels (or ranges) for the number of bikes that should be at each station after rebalancing. We recall that, in our work, the initial inventory level of bikes that should be allocated at stations also acts as target level for the number of bikes to have after rebalancing is performed. Thus, these are decisions for our problem. 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] and Erdoğan et al. [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 the three latter papers, the objective is minimizing rebalancing costs with the use of 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 wherein 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 via 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 the one of Dell'Amico et al. [2018], who propose multiple stochastic programming formulations for dealing 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 users dissatisfaction.

2.3 Stations initial inventory level and rebalancing problems

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 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 a target inventory level for that station is determined. The second step is to determine a dynamic rebalancing plan based upon those target inventory levels, which dictates 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 initial allocation 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] study the problem of both determining the service level requirements (that is, target inventory bounds) at each station and determining 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 initial inventory level at stations to minimize the sum of bikes procurement costs, expected bikes and lockers stockout costs and transshipment costs for performing static rebalancing of bikes. Finally, Ren et al. [2020] study the static rebalancing problem in presence of multiple vehicles and stations which can be visited only 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.

Finally, the presence of a valet service, which allows to expand the capacity of stations by charging an extra penalty, distinguishes our work from the ones in the literature. In fact, to the best of our knowledge, no work in the literature has considered this feature in a bike-sharing initial inventory and rebalancing problem.

Article	Problem features					Solution method
	Fleet size	Initial inventory	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
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

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

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 number of bikes to allocate at the beginning of the day and how they should be rebalanced at a later point during the day, under uncertain demand. The demand at each station is measured as the difference between the uncertain number of rented and returned bikes at that station during the period between when bikes are initially allocated and redistributed. Moreover, the number of bikes to allocate at each station must exceed a minimum threshold to ensure that rental requests early in the day can be satisfied (see Appendix B).

We consider a single depot and multiple stations with given capacities and a number of bikes that are initially allocated. The capacity of each station can be expanded on a temporary basis using the so-called "valet service", which involves individual who can accept returning bikes even when the station is at capacity. We assume that there is a limited number of bikes at the depot corresponding to its capacity that can be allocated to stations. We presume rebalancing is executed by a capacitated vehicle that travels along a known and fixed route (determined *a priori* by solving a Travelling Salesman Problem (TSP)) that begins at the depot, visits each station, and then ends at the depot. Notice that we do not currently 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 total number of bikes initially allocated to stations.

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) are 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 formulation

In this section, we propose a two-stage stochastic programming formulation 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.

We first define the following notation:

Sets:

$\mathcal{I}$: set of bike-stations $\mathcal{I} = \{1, \dots, I\}$ (with I the depot);

$\mathcal{S}$: set of scenarios $\mathcal{S} = \{1, \dots, S\}$ or finite set of possibile realization of the uncertainty;

Deterministic parameters:

$\underline{x}_i$, minimum number of bikes that has to be allocated to station $i \in \mathcal{I} \setminus \{I\}$;

$\overline{I}_{I0}$, 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, delivery penalty at station $i \in \mathcal{I} \setminus \{I\}$;

$t_{i,i+1}$, rebalancing penalty at station $i \in \mathcal{I} \setminus \{I\}$;

Stochastic parameters:

Let $(\Xi, \mathcal{A}, pr)$ be a probability space with Ξ the set of outcomes, σ-algebra $\mathcal{A}$, probability pr and $d \in \Xi$ a particular outcome representing the demand of bike stations. We define:

- d_i^s, stochastic demand of bikes at station $i \in \mathcal{I} \setminus \{I\}$ in scenario $s \in \mathcal{S}$;

- pr^s, probability of scenario $s \in \mathcal{S}$.

Recalling that the demand at each station is measured as the difference between the number of rented and returned bikes at that station during the period between when bikes are initially allocated and redistributed, d_i^s may be either positive (more bikes withdrawn from station i than returned in scenario s) or negative (more bikes returned to station i than withdrawn in scenario s). As highlighted in Raviv and Kolka [2013], this choice of computing demand yields to a "steady-state" demand and ignores the dynamics of the system. In Appendix B, we describe this limitation in more detail and propose a method to counteract it.

We now introduce the decision variables. The first-stage variables are defined as follows:

- $x_i \in \mathbb{Z}^+$: number of bikes to allocate at station $i \in \mathcal{I} \setminus \{I\}$ at the beginning of the service. The decision must be taken before the realizations of the random demand d_i^s.

After the allocation of bikes, at a later point during the day, the stochastic demands d_i^s occur on each station i and the service provider determines a rebalancing plan based on the number of bikes available at each station. Then, the surplus or shortage can be immediately computed at each bike station.

The second-stage decision variables are defined as follows:

- $y_{i,i+1}^s \in \mathbb{Z}^+$, number of bikes to relocate from station $i \in \mathcal{I} \setminus \{I\}$ to station $i+1$ in scenario $s \in \mathcal{S}$;

- $I_i^s \in \mathbb{Z}$, balance of bikes at station $i \in \mathcal{I} \setminus \{I\}$ in scenario $s \in \mathcal{S}$;

- $I_i^{s+} \in \mathbb{Z}^+$, units of surplus at station $i \in \mathcal{I} \setminus \{I\}$ in scenario $s \in \mathcal{S}$;

- $I_i^{s-} \in \mathbb{Z}^-$, units of stock-out at station $i \in \mathcal{I} \setminus \{I\}$ in scenario $s \in \mathcal{S}$;
- $B_i^s \in \mathbb{Z}$, extra inventory balance at station $i \in \mathcal{I} \setminus \{I\}$ in scenario $s \in \mathcal{S}$;
- $B_i^{s+} \in \mathbb{Z}^+$, units of extra inventory at station $i \in \mathcal{I} \setminus \{I\}$ in scenario $s \in \mathcal{S}$;
- $E_i^s \in \mathbb{Z}$, excess inventory balance at station $i \in \mathcal{I} \setminus \{I\}$ in scenario $s \in \mathcal{S}$;
- $E_i^{s+} \in \mathbb{Z}^+$, units of excess inventory 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 Figure 2. The left-most part of Figure 2 represents the first-stage decision, in which bikes are allocated to stations (see the blue arrows) at the beginning of the day, before knowing the realized demand. In the central part of the figure, each orange arrow represents one scenario describing the bike demand at stations. Finally, in the right-most part of Figure 2, 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 redistributed by a vehicle following a fixed route described by the green arrows.

We recall that our problem includes multiple objectives. Specifically, 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+}, representing 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," $\frac{c_i}{Q_i}, \ \forall i \in \mathcal{I} \setminus \{I\}$.

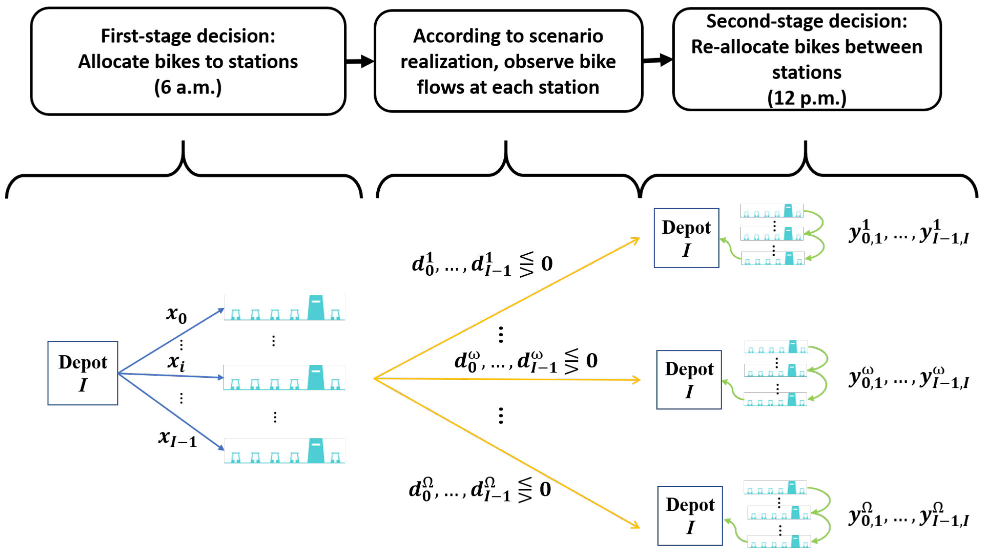


Figure 2: Illustration of the two-stage decision-making process

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, $i \in \mathcal{I} \setminus \{I\}$.

The problem can be formulated as the following integer non linear stochastic program, which we linearize in Appendix A:

Problem $\mathcal{B}$

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

s.t:

$$x_i \geq \underline{x}_i \qquad i \in \mathcal{I} \setminus \{I\} \tag{2}$$

$$\overline{I}_{i0} + x_i \leq Q_i \qquad i \in \mathcal{I} \setminus \{I\} \tag{3}$$

$$\sum_{i \in \mathcal{I} \setminus \{I\}} x_i \leq \overline{I}_{I0} \tag{4}$$

$$y^s_{i,i+1} \leq C \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S} \tag{5}$$

$$I^s_I = \overline{I}_{I0} - \sum_{i \in \mathcal{I} \setminus \{I\}} x_i + y^s_{I-1,I} \qquad s \in \mathcal{S} \tag{6}$$

$$I^s_I \leq \overline{I}_{I0} \qquad s \in \mathcal{S} \tag{7}$$

$$I^s_1 = \overline{I}_{i0} + x_1 - d^s_1 - y^s_{1,2} \qquad s \in \mathcal{S} \tag{8}$$

$$I^s_i = \overline{I}_{i0} + x_i - d^s_i + y^s_{i-1,i} - y^s_{i,i+1} \qquad i \in \mathcal{I} \setminus \{1, I\},\ s \in \mathcal{S} \tag{9}$$

$$I^{s+}_i = \max\{0, I^s_i\} \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S} \tag{10}$$

$$I^{s-}_i = \min\{0, I^s_i\} \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S} \tag{11}$$

$$E^s_i = I^{s+}_i - Q_i \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S} \tag{12}$$

$$E^{s+}_i = \max\{0, E^s_i\} \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S} \tag{13}$$

$$B^s_i = I^{s+}_i - x_i - \overline{I}_{i0} - E^{s+}_i \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S} \tag{14}$$

$$B^{s+}_i = \max\{0, B^s_i\} \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S} \tag{15}$$

$$x_i \geq 0 \ \ \mathit{integer} \qquad i \in \mathcal{I} \setminus \{I\} \tag{16}$$

$$y^s_{i,i+1} \geq 0 \ \ \mathit{integer} \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S} \tag{17}$$

$$I^s_i \ \ \mathit{free\ and\ integer} \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S} \tag{18}$$

$$I^{s+}_i \geq 0 \ \ \mathit{integer} \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S} \tag{19}$$

$$I^{s-}_i \leq 0 \ \ \mathit{integer} \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S} \tag{20}$$

$$B^s_i \ \ \mathit{free\ and\ integer} \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S} \tag{21}$$

$$B^{s+}_i \geq 0 \ \ \mathit{integer} \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S} \tag{22}$$

$$E^s_i \ \ \mathit{free\ and\ integer} \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S} \tag{23}$$

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

The objective function (1) represents the minimization of the expected total cost obtained through the sum of delivery cost for the allocated bikes, the expected rebalancing cost between stations, the expected extra and excess costs and expected stock-out cost for shortage. Constraints (2) impose that the delivered quantity to each station has to be at least as great as the initial requirement at that station. Constraints (3) guarantee that the sum between the quantity allocated and initially available at each station does not exceed the station capacity. Constraint (4) implies that the total number of delivered bikes to stations is less than the available quantity at the depot. Constraints (5) 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. As such, constraints (6) 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 (7) 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 (8) 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 (9) 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 (10) and (11) determine the surplus and stock-out quantities, respectively, for each station and for each scenario $s \in \mathcal{S}$. Constraints (12) and (13) 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 (14) and (15) 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 (16) to (24) define the integrality and non-negativity of first-stage and second-stage variables.

5 Numerical Results

In this section, we present and analyze the results of our computational study with the aim of answering the questions posed in Section 1.

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 7.5.1. All Gurobi parameters were set to their default values.

5.1 Instances and scenario generation

Our computational study is based on the bike-sharing system of the city of San Francisco, CA. This service started in August 2013, and in August 2016 counted 33 bike stations (see Figure 3) and a total of 350 bikes. Open ridership data are available at the web-site www.bayareabikeshare.com/open-data (last accessed in December 2017).

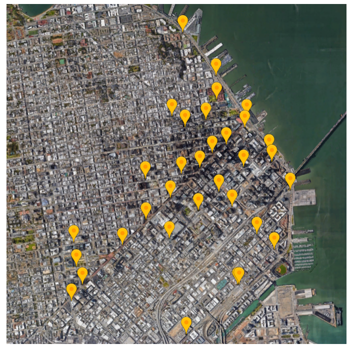


Image created with Google Earth

Figure 3: San Francisco bike stations in August 2016

The parameters of the considered instance are retrieved as follows:

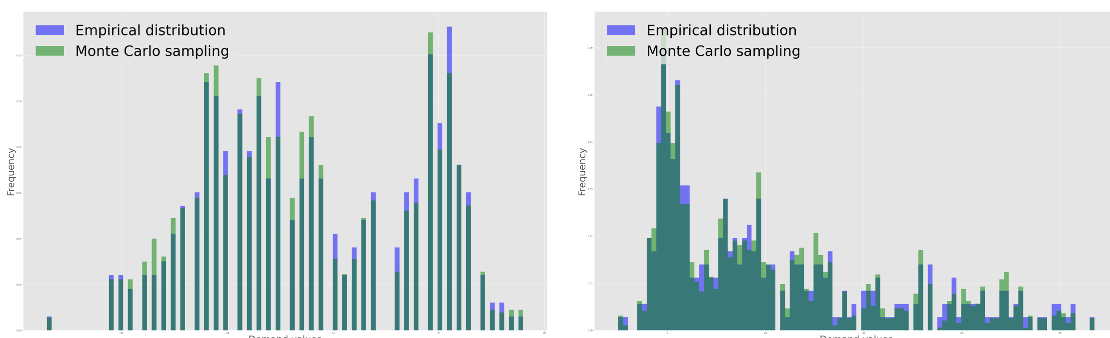
- Minimum number of bikes that has to be allocated to station $\underline{x}_i$, $i \in \mathcal{I} \setminus \{I\}$: see Appendix B;
- Depot capacity $\overline{I}_{I0} = 350$ (value retrieved from our first access to the website);
- Bike-station capacity Q_i, $i \in \mathcal{I} \setminus \{I\}$: see Table 7 in Appendix C;
- Initial availability of bikes $\overline{I}_{i0}$, at station $i \in \mathcal{I} \setminus \{I\}$: see Table 7 in Appendix C;
- Delivery penalty $f_i = 1$ (Dollar), $i \in \mathcal{I} \setminus \{I\}$;
- Rebalancing penalty $t_{i,i+1} = 2$ (Dollars), $i \in \mathcal{I} \setminus \{I\}$;
- Capacity of the vehicle used for transshipment $C = 25$ (see Forma et al. [2015]);
- Excess and stock-out penalties $c_i = p_i = \kappa(1 + \min_{j \in \mathcal{I}: j \neq i} \delta_{ij})$, $i \in \mathcal{I} \setminus \{I\}$, where δ_{ij} is the distance between station i and j calculated as the geodesic distance using the great-circle distance formula (Banerjee [2005]);

Notice that, the allocation penalty f_i and the rebalancing penalty $t_{i,i+1}$ are set to the same values for all stations, i.e., they are not station-dependent. On the contrary, we penalize congestion and starvation by modeling that when a user cannot return (withdraw) a bike to (from) the station they desire, they will instead walk to the next-closest station. Specifically, in the excess and stock-out penalties, considering that the distance to the closest station is typically below one kilometer, we set κ to a high value ($\kappa = 46$) so that, in most of cases, the model redistributes bikes from a station when there is extra inventory to one of the following stations.

5.1.1 Scenario generation

In this section, we aim at answering our first question: How to generate realistic demand scenarios? Scenario generation is an important part of the modelling process, since a bad scenario tree can lead to a solution of the optimization problem that is not meaningful. 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 variable, given by the demand at each station, has a finite number of possible outcomes at the end of the considered period, assumed to be exogenous to the problem. Consequently, the probability distribution is not influenced as well by decisions. Making these assumptions, we can represent the stochastic demand d_i^s, $i \in \mathcal{I}$, $s \in \mathcal{S}$, using a scenario tree which contains a root and a finite set of leaves. In the problem under consideration, the random vector is high-dimensional, and presents complicated dependencies among stations. These factors make the uncertainty very difficult to represent. For this reason, we derived an empirical distribution of each station demand as inverse of the Kaplan-Meier estimate of the cumulative distribution function (also known as the empirical cdf) of the real historical ridership data that we collected and that we denote as d_i^n, $i \in \mathcal{I} \setminus \{I\}, n = 1, \ldots, N$, where N is the total number of collected data. Specifically, for each day and each station, we computed the number of withdrawn and returned bikes between 6 am and 11:59 am, and set the demand for that day as the difference between those two numbers (withdrawn-returned). Since we add penalties in the objective function for extra and excess inventory and for stockout, we do not bound the demand values to stations capacities.

From the empirical demand distributions, several scenario trees of increasing size are then generated according to a Monte Carlo sampling procedure. Pseudo-code (1) presents the details for scenario generation. Finally, Figures 4a and 4b illustrate for two chosen stations in San Francisco a comparison of the empirical distribution based on 369 real observations and Monte Carlo sampling based on 2,000 observations. For the sake of representation, we display their relative frequencies for the two distributions to be on the same scale. Results show that the two patterns follow a similar behaviour.



(a) Station 1

(b) Station 2

Figure 4: Empirical distributions and Monte Carlo sampling for two different stations

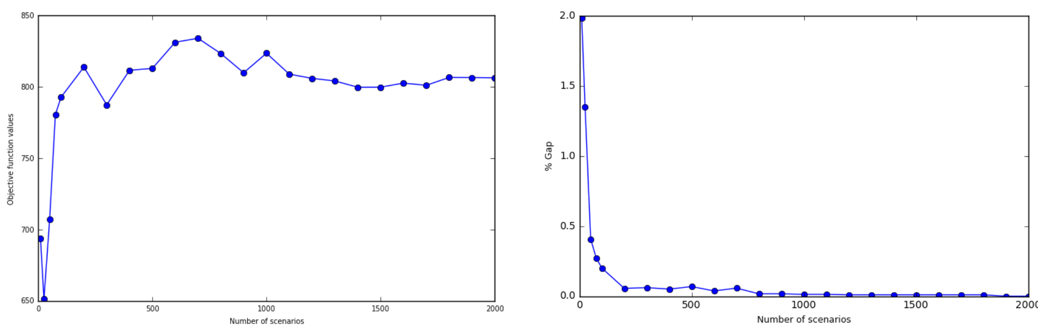
Pseudo-code 1: Scenario generation process

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

5.1.2 Determining the size of the scenario tree

In this section, we aim at answering our second question: What is the minimum number of scenarios to be considered to accurately describe the stochastic demand?

We performed both an in-sample and out-of-sample stability analysis of our stochastic program, following the procedure described in Kaut and Wallace [2007]. We illustrate the results of these analyses in Figures 5a and 5b. Regarding the in-sample analysis, we solved the stochastic program for scenario trees of increasing size. Figure 5a indicates that the objective function value stabilizes with 1,200 scenarios. However, we have to remember that in-sample values are not directly comparable. To be able to estimate the effect of using a larger scenario tree, we have to compare the out-of-sample costs. For this purpose, we declare a scenario tree with 2,000 scenarios 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 using scenario trees with a smaller size. We see in Figure 5b that convergence of the out-of-sample analysis is nearly monotonic and decreasing with a percentage gap under 0.1%. In the following, we base our computational study on a set of 1200 scenario which we assume to be equi-probable, i.e. we set $pr^s = \frac{1}{|\mathcal{S}|}, \ \ \forall s \in \mathcal{S}$.



(a) In-sample stability analysis results

(b) Out-of-sample stability analysis results

Figure 5: In sample and out-of-sample analysis

5.2 Stochastic solution analysis

In this section, we analyze the value of the stochastic solution (Section 5.2.2), and the quality of the expected value solution (Section 5.2.3). To perform both analyses, we use typical indicators from the literature and a simulation model that we describe in Section

5.2.1.

5.2.1 Simulation model

To simulate the performance of an initial allocation of bikes to station, we consider historical ridership data of the week of June 20, 2016 to June 26, 2016.

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. In doing this, we record statistics related to: (i) the number of times a user wants to return a bike to a station, but it is full ("% congestion"), and (ii) the number of times a user wants to withdraw a bike from a station, but it is empty ("% starvation"). Then, we compute the final inventory level at each station, $I_i^{final} = \overline{I}_{i0} + x_i - w_i + s_i$, 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 stochastic program to determine the number of rebalanced bikes. Similarly, we calculate the total number of rebalanced bike miles ("miles · rebalanced qty"), as $\sum_{i,i+1} \delta_{i,i+1} \bar{y}_{i,i+1}$, where $\bar{y}_{i,i+1}$ indicates how many bikes should be rebalanced from station i to station $i+1$ according to the rebalancing plan. Because observing how the system performs after rebalancing is done is interesting, an alternative perspective on these final inventory levels is to calculate a fill rate-type statistic, wherein we estimate the percentage of future bike withdrawals that can be satisfied. For this, we use the same demand scenario set computed in Section 5.1.2, and compute the fill rate of station i in scenario s as follows:

$$FR_i^s = \begin{cases} 1 & \text{if } I_i^{final} - d_i^s \geq 0 \\ \frac{I_i^{final}}{d_i^s} & \text{otherwise} \end{cases} \tag{25}$$

To get the average expected fill rate, we average this statistic over all stations and scenarios and, in the following, we refer to it with "expected FR". Finally, we compute the total number of "extra inventory qty" that represents the total number of bikes in excess with respect to the initial number at stations. We interpret this statistic as the number of bikes that cannot be redistributed through rebalancing.

5.2.2 Analyzing the value of stochastic solution

In this section, we aim at answering to our third question: What is the Value of the Stochastic Solution (VSS) (see Birge and Louveaux [2011], Kall and Wallace [1994])? To do so, we solve the stochastic program presented above on the benchmark scenario tree, to get its optimal objective function value, RP. We then solve the *Expected Value Problem (EV)*, which is obtained by solving a deterministic version of our stochastic program, in which the random demand parameters are replaced with their expected values, and rounded to the nearest integer (bike demands cannot be given by fractional values). We then evaluate how the deterministic 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 stochastic program and we compute the (Relative) *Value of Stochastic Solution*

$$\%VSS = (EEV - RP)/RP = 41.15\%,$$

suggesting that significant gains can be realized by solving the stochastic program versus the expected value approach.

We also assess how well the initial allocation plans from both the EV problem and the stochastic program perform in the simulation model (see Section 5.2.1), and report the above-mentioned statistics related to the performance of each plan in Table 2. We see that while the stochastic program allocates 40% more bikes, it leads to a much smaller frequency of starvation and higher fill rate. However, despite of the high difference in the number of allocated bikes in the stochastic and in the deterministic solution, the fill rates are similar. There are two reasons for this: (i) the fill rate already includes the effect of rebalancing, so that if the bike redistribution is efficient, potential uncovered withdrawals are corrected, and (ii) the higher number of allocated bikes in the stochastic program can be caused by the need of reducing rebalancing operations. In fact, we observe that the initial plan prescribed by the stochastic program also requires less rebalancing. However, as it allocates more bikes, the plan prescribed by the stochastic solution also yields a higher frequency of congestion, which is limited compared to the higher frequency of starvation of the deterministic solution.

Table 2: Simulation-based comparison of solutions to stochastic and expected value problems.

	SP					EEV				
Day	% congestion	% starvation	miles · rebalanced qty	expected FR	extra inventory qty	% congestion	% starvation	miles · rebalanced qty	expected FR	extra inventory qty
Mon	8.81%	31.30%	19.95	85.86%	149	5.56%	39.57%	50.65	86.28%	151
Tue	9.21%	28.74%	22.17	86.28%	149	6.53%	38.32%	49.70	86.45%	153
Wed	7.27%	31.11%	16.85	87.55%	158	4.44%	40.50%	46.55	88.13%	163
Thu	6.94%	27.88%	15.34	86.43%	141	4.49%	36.01%	42.78	86.17%	145
Fri	6.68%	25.98%	19.89	86.42%	105	3.23%	36.55%	51.59	87.03%	109
Sat	7.62%	21.36%	3.36	88.22%	37	0.00%	23.30%	3.36	85.26%	37
Sun	0.00%	20.83%	2.31	91.32%	19	0.00%	25.00%	2.31	88.04%	19
Average	6.65%	26.74%	14.27	87.44%	108.29	3.46%	34.18%	35.28	86.77%	111
Tot delivered bikes	154					110				

5.2.3 Analyzing the quality of the Expected Value solution

In this section, we aim at answering our fourth question: What is the quality of the Expected Value (EV) solution when solving the stochastic model? To do so, we compute two more indicators: the *Loss of Using the Skeleton Solution (LUSS)* and the *Loss of Upgrading the Deterministic Solution (LUDS)* defined in Maggioni and Wallace [2012].

We interpret the skeleton solution as the stations to which exactly the minimum number of required bikes is allocated. To compute the *LUSS*, we examine the solution to the EV problem to determine the subset of stations $\bar{\mathcal{I}}$ to which it allocates exactly the minimum number of bikes $\underline{x}_i$. We then solve the stochastic program, fixing $x_i = \underline{x}_i$ 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 found the (Relative) *LUSS* measure,

$$\%LUSS = (ESSV - RP)/RP = 7.95\%.$$

The positive $LUSS$ value means that the expected value solution selects the wrong quantities to deliver to the wrong stations and its structure (skeleton) cannot be inherited in a stochastic environment. Relatively, we illustrate in Figure 6a the stations in $\mathcal{I} \setminus \bar{\mathcal{I}}$ in the solution to the EV problem and in Figure 6b the analogously-determined stations in the solution to the stochastic program *SP*. From the figures we conclude that the solution of the EV model allocates bikes to too few stations compared to the *SP* one. Specifically, in the EV solution only three stations receive a higher number of bikes than the initial requirement, compared to the *SP* in which these stations are twelve. We also note that the three stations receiving a higher number of bikes than the initial requirement in the EV solution show the same behavior in the *SP* solution.

(a) First-stage EV skeleton solution (b) First-stage SP skeleton solution

Images created with Google Earth

Figure 6: Distributions of bikes to stations in the EV and SP solutions

With the $LUDS$, we seek to determine whether the solution to the EV problem is upgradeable, i.e., that it can be used as a starting point for generating a high-quality solution to the stochastic program. To do so, we solve the stochastic program, 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 EV problem. We refer to the objective function value of the optimal solution to this restricted stochastic program as the *Expected Input Value (EIV)* and compute a relative $LUDS$ measure as

$$\%LUDS = (EIV - RP)/RP = 0\%.$$

The result suggests that the EV solution is *perfectly upgradeable*, indicating that solving the EV problem can be a good start for solving the stochastic program. Besides, we obtained that by first solving the EV problem, and then the LUDS-restricted stochastic program, the total solution time is reduced by 10% of what it is needed to solve the stochastic program from scratch. This result shows that applying this heuristic procedure that uses information from the deterministic solution is valuable.

5.3 A comparison with the implemented system

In this section, we aim at answering our fifth question: How does the actual allocation plan implemented in San Francisco perform compared to the ones obtained by optimally solving the stochastic and deterministic models? We finish with a comparison of how the plan prescribed by the stochastic program performs relative to what we determined was the initial allocation plan in the actual system. We derived the actual allocation of bikes to stations from station status data by collecting the number of bikes at each station at 6 a.m., for each day of the considered week. We compare the performance of the *RP* initial allocation plan with the simulation model described in Section 5.2.1. We present statistics regarding each plan in Table 3. We see that the stochastic program allocated far fewer bikes (45% fewer), which in turn lead to less congestion and rebalancing relative to the actual San Francisco service. However, not surprisingly, we saw an increase in the starvation frequency.

Table 3: Comparison of the actual allocation plan with the one obtained from the stochastic program

	Actual allocation plan						SP allocation plan					
Day	total bikes at 6 a.m.	% congestion	% starvation	miles · rebalanced qty	expected FR	extra inventory qty	total bikes at 6 a.m.	% congestion	% starvation	miles · rebalanced qty	expected FR	extra inventory qty
Mon	272	17.24%	25.39%	40.09	90.92%	113	155	8.81%	31.30%	19.95	85.86%	149
Tue	275	14.97%	19.76%	38.19	88.58%	109	155	9.21%	28.74%	22.17	86.28%	149
Wed	281	15.76%	23.38%	24.40	91.49%	118	155	7.47%	31.11%	16.85	87.55%	157
Thu	289	13.88%	22.22%	38.76	90.92%	101	155	6.94%	27.88%	15.34	86.43%	141
Fri	285	10.14%	14.94%	22.73	91.46%	85	155	6.68%	25.98%	19.89	86.42%	104
Sat	293	4.76%	2.91%	3.36	94.03%	31	155	7.62%	21.36%	3.36	88.26%	37
Sun	288	17.39%	14.58%	2.12	92.80%	18	155	0.00%	20.83%	2.31	91.34%	19
Average	283.29	13.45%	17.58%	24.24	91.46%	82.14	155	6.68%	26.74%	14.27	85.45%	108

In the following, to normalize our comparison, we determine the initial allocation of bikes to stations by solving a stochastic program that determines how many bikes to allocate to each station but under the constraint that the total number of allocated bikes has to be equal to the one of the real system. We present the related statistics in Table 4. We observe that the allocation plans prescribed by the stochastic program outperform the actual allocation on each day and in each category. These results represent a strong indicator of the impact that the proposed stochastic approach could have in practice.

Table 4: Comparison of implemented plan and plan from stochastic program, when allocating the same number of bikes.

Day	Actual allocation plan						SP with total number of allocated bikes of the actual allocation plan					
	total bikes at 6 a.m.	% congestion	% starvation	miles · rebalanced qty	expected FR	extra inventory qty	total bikes at 6 a.m.	% congestion	% starvation	miles · rebalanced qty	expected FR	extra inventory qty
Mon	272	17.24%	25.39%	40.09	90.92%	113	272	12.07%	18.70%	28.21	91.92%	127
Tue	275	14.97%	19.76%	38.19	88.58%	109	275	11.90%	18.56%	37.73	92.09%	117
Wed	281	15.76%	23.38%	24.40	91.49%	118	281	9.70%	22.97%	24.40	93.46%	140
Thu	289	13.88%	22.22%	38.76	90.92%	101	289	9.18%	16.98%	24.43	93.64%	114
Fri	285	10.14%	14.94%	22.73	91.46%	85	285	7.83%	13.79%	20.20	93.54%	93
Sat	293	4.76%	2.91%	3.36	94.03%	31	293	7.62%	2.91%	4.16	96.90%	33
Sun	288	17.39%	14.58%	2.12	92.80%	18	288	4.35%	8.33%	2.31	97.80%	19
Average		13.45%	17.58%	24.24	91.46%	82.14		8.95%	14.61%	20.21	94.19%	91.86

To gain more insights on the allocation of the bikes, in Table 5, we report the standard deviation, the minimum, and the maximum of the number of allocated bikes in the actual allocation plan and in the solution of our stochastic program. The statistics suggest that the stochastic program allocates always at least one bike to every station, while, in the actual allocation plan, at least one station has no bikes allocated. Moreover, the difference of the number of allocated bikes across stations is more pronounced in the stochastic solution as shown by the higher standard deviation.

Table 5: Standard deviation, minimum, and maximum of the number of allocated bikes in the actual allocation plan and in the solution of our stochastic program.

	Actual allocation plan			SP with total number of allocated bikes of the actual allocation plan		
Day	std. dev	min	max	std. dev	min	max
Mon	5.18	2	19	6.06	1	23
Tue	5.95	1	21	6.06	1	23
Wed	6.30	1	21	6.17	1	23
Thu	5.12	1	21	6.14	1	23
Fri	5.20	0	20	6.15	1	23
Sat	5.79	0	23	6.12	1	23
Sun	5.90	1	22	6.14	1	23

6 Conclusion and future work

In this paper, we studied the problem of determining an initial allocation of bikes to stations, as well as 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 this allocation is that there are multiple dimensions along which the performance of such an allocation plan can be measured, with some measuring costs incurred while operating the system and others measuring the quality of the service experienced by users of the system. As a result, we present a two-stage stochastic program wherein the first-stage variables determine this initial allocation of bikes, and the second-stage variables determine how bikes are rebalanced at a point later in the day. Another challenge in this setting is determining how to measure demand, as, like a rental system, bikes are both withdrawn and returned from individual stations.

We performed a computational study based upon historical ridership data from the bike-sharing system of the city of San Francisco. In particular, we used this data to derive a simulation model wherein we can estimate the performance of an initial allocation plan along multiple dimensions. With that study, we first 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. 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 initial allocation plan prescribed by the stochastic program with what we estimated was the initial allocation plan for a given week of historical data. We saw that the stochastic program produced a much better initial allocation of bikes than what we estimated was done in practice. In addition, our solution suggested to allocate fewer bikes than the ones in the real system. Consequently, practitioners should carefully evaluate whether to allocate the total number of available bikes at the depot to reduce rebalancing and risks of congestion.

Regarding future work, we believe the next logical step in this research is to consider a multi-stage stochastic optimization model 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 horizons approaches (see Bertazzi and Maggioni [2018]). Finally, another extension is to model that multiple vehicles can be used to support rebalancing and that their routes can change from one day to the next.

Acknowledgments

The authors thank the Transportation and Logistics Society (TSL) as this research has been supported in part by the TSL Cross Region Doctoral Grant.

References

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

Sudipto Banerjee. On geodetic distance computations in spatial modeling. *Biometrics*, 61 (2):617–625, 2005.

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

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

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

John R Birge and François Louveaux. *Introduction to stochastic programming.* Springer Science & Business Media, 2011.

Jan Brinkmann, Marlin W Ulmer, and Dirk C Mattfeld. Short-term strategies for stochastic inventory routing in bike sharing systems. *Transportation Research Procedia*, 10: 364–373, 2015.

Jan Brinkmann, Marlin W Ulmer, and Dirk C Mattfeld. Inventory routing for bike sharing systems. *Transportation Research Procedia*, 19:316–327, 2016.

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

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

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

Teobaldo Bulhões, Anand Subramanian, Güneş Erdoğan, and Gilbert Laporte. The static bike relocation problem with multiple vehicles and visits. *European Journal of Operational Research*, 264(2):508–523, 2018.

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

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

Fábio Cruz, Anand Subramanian, Bruno P Bruck, and Manuel Iori. A heuristic algorithm for a single vehicle static bike sharing rebalancing problem. *Computers & Operations Research*, 79:19–33, 2017.

Sharon Datner, Tal Raviv, Michal Tzur, and Daniel Chemla. Setting inventory levels in a bike sharing network. *Transportation Science*, 2017.

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

Mauro Dell'Amico, Manuel Iori, Stefano Novellani, Thomas Stützle, et al. A destroy and repair algorithm for the bike sharing rebalancing problem. *Computers & Operations Research*, 71:149–162, 2016.

Mauro Dell'Amico, Manuel Iori, Stefano Novellani, and Anand Subramanian. The bike sharing rebalancing problem with stochastic demands. *Transportation Research Part B: Methodological*, 118:362–380, 2018.

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

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

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

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

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

Sin C Ho and WY Szeto. Solving a static repositioning problem in bike-sharing systems using iterated tabu search. *Transportation Research Part E: Logistics and Transportation Review*, 69:180–198, 2014.

Peter Kall and Stein W Wallace. *Stochastic Programming.* Springer, 1994.

Michal Kaut and Stein W Wallace. Evaluation of scenario-generation methods for stochastic programming. *Pacific Journal of Optimization*, 3(2):257–271, 2007.

Alan J King and Stein W Wallace. *Modeling with stochastic programming.* Springer Science & Business Media, 2012.

Benjamin Legros. Dynamic repositioning strategy in a bike-sharing system; how to prioritize and how to rebalance a bike station. *European Journal of Operational Research*, 272(2):740–753, 2019.

Chung-Cheng Lu. Robust multi-period fleet allocation models for bike-sharing systems. *Networks and Spatial Economics*, 16(1):61–82, 2016.

Chang Lv, Chaoyong Zhang, Kunlei Lian, Yaping Ren, and Leilei Meng. A hybrid algorithm for the static bike-sharing re-positioning problem based on an effective clustering strategy. *Transportation Research Part B: Methodological*, 140:1–21, 2020.

Francesca Maggioni and Stein W Wallace. Analyzing the quality of the expected value solution in stochastic programming. *Annals of Operations Research*, 200(1):37–54, 2012.

Francesca Maggioni, Matteo Cagnolari, Luca Bertazzi, and Stein W Wallace. Stochastic optimization models for a bike-sharing problem with transshipment. *European Journal of Operational Research*, 276(1):272–283, 2019.

Rahul Nair and Elise Miller-Hooks. Fleet management for vehicle sharing operations. *Transportation Science*, 45(4):524–540, 2011.

Bruno A Neumann-Saavedra, Patrick Vogel, and Dirk C Mattfeld. Anticipatory service network design of bike sharing systems. *Transportation Research Procedia*, 10:355–363, 2015.

PBSC. https://www.pbsc.com/blog/2021/10/the-meddin-bike-sharing-world-map, 2021.

Tal Raviv and Ofer Kolka. Optimal inventory management of a bike-sharing station. *IIE Transactions*, 45(10):1077–1093, 2013.

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

Robert Regue and Will Recker. Proactive vehicle routing with inferred demand to solve the bikesharing rebalancing problem. *Transportation Research Part E: Logistics and Transportation Review*, 72:192–209, 2014.

Yaping Ren, Leilei Meng, Fu Zhao, Chaoyong Zhang, Hongfei Guo, Ying Tian, Wen Tong, and John W Sutherland. An improved general variable neighborhood search for a static bike-sharing rebalancing problem considering the depot inventory. *Expert Systems with Applications*, 160:113752, 2020.

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

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

Patrick Vogel. Service network design of bike sharing systems. In *Service Network Design of Bike Sharing Systems*, pages 113–135. Springer, 2016.

Patrick Vogel, Bruno A Neumann Saavedra, and Dirk C Mattfeld. A hybrid metaheuristic to solve the resource allocation problem in bike sharing systems. In *International Workshop on Hybrid Metaheuristics*, pages 16–29. Springer, 2014.

Appendix A Model linearization

In this section, we present the linearization of the model presented in Section 4. First, we need to modify the objective function (1) as follows:

$$min \sum_{i \in \mathcal{I}} f_i x_i + \sum_{s \in \mathcal{S}} pr^s [\sum_{i \in \mathcal{I} \setminus \{I\}} (t_{i,i+1} y^s_{i,i+1} + \frac{c_i}{Q_i} B^{s+}_i + c_i E^{s+}_i + p_i I^{s-}_i)]. \tag{26}$$

Then, we need to linearize the expressions for determining I^{s+}_i and I^{s-}_i. We introduce the binary variable z^s_i such that:

$$z^s_i = \begin{cases} 1 & \text{if } I^s_i \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

and we substitute constraints (10) and (11) with:

$$I^{s+}_i \geq I^s_i \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S} \tag{27}$$

$$I^{s+}_i \leq I^s_i + \max_{s \in \mathcal{S}}(d^s_i)(1 - z^s_i) \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S} \tag{28}$$

$$I^{s+}_i \geq 0 \ \ integer \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S} \tag{29}$$

$$I^{s+}_i \leq M z^s_i \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S} \tag{30}$$

$$I^{s-}_i \geq -I^s_i \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S} \tag{31}$$

$$I^{s-}_i \leq -I^s_i + M z^s_i \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S} \tag{32}$$

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

$$I^{s-}_i \leq \max_{s \in \mathcal{S}}(d^s_i)(1 - z^s_i) \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S} \tag{34}$$

The same linearization technique is applied for E^{s+}_i, and B^{s+}_i. In particular, we introduce the binary variables e^s_i and r^s_i, such that:

$$e^s_i = \begin{cases} 1 & \text{if } E^s_i \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

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

and we substitute constraints (13) and (15) with the following:

$$B_i^{s+} \geq B_i^s \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S} \tag{35}$$

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

$$B_i^{s+} \leq M r_i^s \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S} \tag{37}$$

$$B_i^{s+} \geq 0 \;\; \textit{integer} \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S} \tag{38}$$

$$B_i^{s-} \geq -B_i^s \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S} \tag{39}$$

$$B_i^{s-} \leq -B_i^s + M r_i^s \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S} \tag{40}$$

$$B_i^{s-} \leq M(1 - r_i^s) \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S} \tag{41}$$

$$B_i^{s-} \geq 0 \;\; \textit{integer} \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S} \tag{42}$$

$$E_i^{s+} \geq E_i^s \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S} \tag{43}$$

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

$$E_i^{s+} \leq M e_i^s \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S} \tag{45}$$

$$E_i^{s+} \geq 0 \;\; \textit{integer} \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S} \tag{46}$$

$$E_i^{s-} \geq -E_i^s \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S} \tag{47}$$

$$E_i^{s-} \leq -E_i^s + M e_i^s \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S} \tag{48}$$

$$E_i^{s-} \leq M(1 - e_i^s) \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S} \tag{49}$$

$$E_i^{s-} \geq 0 \;\; \textit{integer} \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S} \tag{50}$$

Finally, we introduce the variable definition constraints:

$$z_i^s \in \{0,1\} \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S} \tag{51}$$

$$r_i^s \in \{0,1\} \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S} \tag{52}$$

$$e_i^s \in \{0,1\} \qquad i \in \mathcal{I} \setminus \{I\},\ s \in \mathcal{S} \tag{53}$$

Appendix B Determining the initial bike requirement for each station

As highlighted in Raviv and Kolka [2013], computing demand as the difference between the number of withdrawn and returned bikes during a time interval yields to a "steady-state" demand and ignores the dynamics of the system. As an example, consider a station from which a bike is withdrawn and then, one is returned, and no other bikes are withdrawn or returned. The resulting station demand would be counted as zero, indicating to the model that no bikes need to be allocated to that station. With zero bikes allocated, however, the first withdrawal request cannot be satisfied. We counteract to this limitation by estimating the number of bikes withdrawn from a station before any are returned and defining this as the minimum number of bikes that has to be allocated to a station.

To ensure each station has a sufficient number of bikes to satisfy rental requests early in the day, before any bikes are returned to that station, we determine an initial bike requirement for each station. We consider five methods for estimating the number of withdrawn bikes before a return occurs, and test their relative performance with our simulation. All are based on calculating statistics from historical ridership data over the time period from 6 am until 11:59 am. These methods are based on one of the following statistics for each station: (1) Average bike-interarrival time for bikes returned to that station, (2) The average trip time for bikes returned to that station, (3) The average time until the first return to that station, and, (4) The total number of bikes withdrawn from that station between 6 am and 11:59 am. The methods are as follows:

- Method 1: We estimate the initial requirement as the average number of withdrawn bikes between 6 a.m. and 6 a.m., plus the average bike inter-arrival time.
- Method 2: We estimate the initial requirement as the average number of withdrawn bikes between 6 a.m. and 6 a.m., plus thie average trip time.
- Method 3: We estimate the initial requirement as this total number of withdrawn bikes divided by the average bike inter-arrival time.
- Method 4: We estimate the initial requirement as this total number of withdrawn bikes divided by the average trip time.
- Method 5: We estimate the initial requirement as the total number of withdrawn bikes between 6 a.m. and 6 a.m., plus the average time until first return.

To assess each method, we first solve the stochastic program, while requiring that the number of bikes initially allocated to each station is at least as great as the number suggested by that method. We then run our simulation, with those initial allocations, and compute the frequency of congestion (starvation) relative to the number of bikes returned (withdrawn). We report these relative frequencies in Table 6. There, we see that methods 3 and 5 perform the best with respect to starvation, with 5 performing slightly better with respect to congestion. We also see that fewer bikes are allocated with method 5. Thus, we conclude that method 5 is the best method, and used it for the remainder of our computational study.

Table 6: Congestion and starvation relative frequencies by method of determining initial bike requirement

	Method 1		Method 2		Method 3		Method 4		Method 5	
Day	% congestion	% starvation	% congestion	% starvation	% congestion	% starvation	% congestion	% starvation	% congestion	% starvation
Mon	8.43%	33.07%	8.43%	33.07%	9.00%	29.53%	8.43%	32.68%	8.81%	31.30%
Tue	8.64%	30.94%	8.64%	31.14%	9.41%	28.74%	8.64%	30.54%	9.21%	28.74%
Wed	7.07%	32.57%	7.07%	32.57%	7.07%	30.48%	7.07%	32.57%	7.27%	31.11%
Thu	6.94%	30.19%	6.94%	30.19%	7.14%	28.30%	6.94%	30.19%	6.94%	27.88%
Fri	6.68%	28.97%	6.68%	28.97%	6.91%	25.29%	6.68%	28.97%	6.68%	25.98%
Sat	7.62%	32.04%	7.62%	33.01%	7.62%	17.48%	7.62%	30.10%	7.62%	21.36%
Sun	0.00%	29.17%	0.00%	31.25%	0.00%	20.83%	0.00%	31.25%	0.00%	20.83%
Average	**6.48%**	**30.99%**	**6.48%**	**31.46%**	**6.74%**	**25.81%**	**6.48%**	**30.90%**	**6.65%**	**26.74%**
Initial requirement	55		36		103		57		89	
Total delivery	147		146		160		148		154	

Appendix C Station capacities and initial availabilities

Station id	Station name	Q_i	$\overline{I}_{i0}$
1	Townsend at 7th	15	3
2	San Francisco Caltrain 2 (330 Townsend)	23	5
3	San Francisco Caltrain (Townsend at 4th)	19	8
4	2nd at Townsend	27	2
5	2nd at South Park	15	3
6	2nd at Folsom	19	2
7	Howard at 2nd	19	1
8	Temporary Transbay Terminal (Howard at Beale)	23	8
9	Embarcadero at Bryant	15	4
10	Spear at Folsom	19	2
11	Embarcadero at Folsom	19	2
12	Steuart at Market	23	2
13	Harry Bridges Plaza (Ferry Building)	23	4
14	Davis at Jackson	15	2
15	Embarcadero at Vallejo	15	1
16	Embarcadero at Sansome	15	2
17	Washington at Kearney	15	2
18	Commercial at Montgomery	15	1
19	Clay at Battery	15	1
20	Beale at Market	19	2
21	Mechanics Plaza (Market at Battery)	19	1
22	Market at Sansome	27	2
23	Post at Kearney	19	1
24	Powell at Post (Union Square)	19	5
25	Market at 4th	19	1
26	Yerba Buena Center of the Arts (3rd at Howard)	19	1
27	5th at Howard	15	2
28	Powell Street BART	19	3
29	Civic Center BART (7th at Market)	23	4
30	Golden Gate at Polk	23	3
31	San Francisco City Hall	19	3
32	Market at 10th	27	4
33	South Van Ness at Market	19	2

Table 7: Station capacities and bike initial availability levels.