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

Rossana Cavagnini\textsuperscript{1}, Luca Bertazzi\textsuperscript{2}, Francesca Maggioni\textsuperscript{1}, and Mike Hewitt\textsuperscript{3}

\textsuperscript{1}Department of Management, Economics and Quantitative Methods, University of Bergamo, Italy
\textsuperscript{2}Department of Economics and Management, University of Brescia, Italy
\textsuperscript{3}Department of Information Systems and Operations Management, Quinlan School of Business, Loyola University Chicago, USA

r.cavagnini@studenti.unibg.it, luca.bertazzi@unibs.it, francesca.maggioni@unibg.it, mhewitt3@luc.edu

May 22, 2018

Abstract

The Bikesharing allocation and rebalancing problem is the problem of determining the initial daily allocation of bikes to stations in a bikesharing system composed of one depot and multiple capitiated stations, in which bikes can be rebalanced at a point in time later in the day. We propose a two-stage stochastic programming formulation, where the allocation is made in the first-stage and the recourse decisions related to rebalancing are made in 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 solution of the deterministic equivalent formulation. Nevertheless, we derive a deterministic solution-based heuristic for solving the stochastic program that significantly reduces its solution time without losing solution quality. We benchmark our approach on the real bikesharing 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 1000 in 2016 (World Economic Forum [2016]). In North America, bike sharing systems can be found in New York, Chicago, Washington, the San Francisco Bay Area, and Boston, whereas in South America, they can be found in Buenos Aires and Rio de Janeiro. In Europe, systems can be found in Copenhagen, London, and Paris. Finally, in Asia, the Wuhan and Hangzhou systems are two of the largest in the world, whereas in Oceania, large bike sharing systems can be found in Brisbane and Melbourne. The prevalence 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 when done. We note that the rider need not return the bike to 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 bikesharing 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 after the realization of the demand, through a capacitated vehicle that follows 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. 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 for allocating and re-distributing bikes, and show that explicitly acknowledging uncertainty 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 the problem solution is examined, showing the benefits of the proposed methodology in the solution quality when compared to solving the deterministic equivalent formulation. Nevertheless, by assessing the upgradeability of solutions from a deterministic version of this problem, we derive a heuristic procedure for solving
the stochastic program that significantly reduces its solution time without losing solution quality.

To evaluate the plan produced by the stochastic program, we execute it in the context of a simulation of one week of operation for the San Francisco bike sharing system. To measure the impact of this plan on user experience, during this simulation we compute the frequency of congestion and starvation events at stations. To measure its impact on penalties, we measure the number of bikes allocated to and reallocated between stations. We see that the stochastic program is superior, as it outperforms the actual plan 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. Next, in Section 3, we describe the problem and, in Section 4, we formulate the associated model. In Section 5, we present computational results and analysis, while Section 6 includes managerial insights. Finally, Section 7 provides conclusions and suggestions for future works.

2 Literature Review

In this section, we review the literature that is relevant to the problem we study. It has five key features: (1) it includes various and contrasting objectives, as it considers both measures of customer service and the penalties associated with providing that service; (2) it is stochastic, as it explicitly recognizes that both the number of bikes rented from and returned to each station is not known with certainty but a probability distribution is given; (3) it captures the opportunity to rebalance bikes at a point in time later in the day; (4) it considers that rebalancing is done with a fixed and static route, as motivated by the advantages highlighted in Erera et al. [2009], such as the regularity of the service, the simplification of the planning process and the familiarization of drivers with the delivery area.
This is a fundamental difference from much of the existing literature, as much of it also considers the opportunity to determine the vehicle route used to support rebalancing each day; (5) it includes the presence of a Valet service, wherein station capacity is augmented by another resource (an individual) who can accept returning bikes. This service has been provided by the bikesharing system of Chicago (www.divvybikes.com/valet). When discussing the relevant literature, we will highlight how that work compares with respect to these features.

A natural categorization of the relevant literature is into problems that recognize uncertainty in bike usage and those that do not. Thus, in subsection 2.1 we first review the literature that considers deterministic problems and then, in subsection 2.2, the literature that studies problems that recognize this source of uncertainty. However, there is a literature on dynamic rebalancing problems, wherein (rebalancing) decisions are made while bikes are in use and thus, in subsection 2.3, we provide also a discussion on that literature.

2.1 Deterministic bikesharing problems

The majority of papers in the literature assumes that the use of bikes (i.e., how many are withdrawn from and returned to each station) is known when allocation and (potentially) rebalancing decisions are made. Raviv and Kolka [2013] propose a deterministic bi-objective inventory model, in which the only decision to take is the initial allocation of bikes to stations. The model penalizes both when a user cannot withdraw a bike from a given station because it has stocked out as well as when a user cannot return a bike to a given station because it is at capacity. The model then seeks to minimize the sum of these penalties. However, these penalties are the same for all stations. In contrast, we propose to base these penalties on the distance between a station and the next closest station to capture that the individual may have to travel to that next closest station to either rent or return a bike. Differently from our work, this model also does not consider the option
to rebalance at a later point in time.

Much of the literature focuses on what is referred to as the static bike rebalancing problem (BRP), which models the rebalancing of bikes at a point in time where there is very little usage (e.g. late at night). Such a problem can be viewed as a static pickup and delivery problem. A review and classification of this family of problems is presented in Berbeglia et al. [2007]. According to their classification, the problem we study is most 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 in which the rebalancing route visits stations. However, in contrast to our problem, this variant of the PDP does not model the customer service aspect of the problem.

Conversely, Raviv et al. [2013] models the static rebalancing problem with a multi-objective mathematical program wherein they seek to minimize an objective that consists of three terms: (1) penalties for stockouts, (2) penalties for stations being at capacity when users wish to return bikes, and, (3) the operational costs incurred when rebalancing. They present two formulations of the problem, with the first restricting how rebalancing can be performed in order to make the model easier to solve. The second formulation then relaxes some of those restrictions. A variant of the first formulation is also studied in Forma et al. [2015] and in Ho and Szeto [2014].

Benchimol et al. [2011], Chemla et al. [2013], and Erdoğan et al. [2014] also focus on the static rebalancing problem. In addition, all three propose models that seek to minimize some measure of the costs incurred by rebalancing. Benchimol et al. [2011] and Chemla et al. [2013] model the desire to avoid starvation and congestion with the use of pre-determined target levels for the number of bikes at each station after rebalancing. Brinkmann et al. [2016] also considers a target level for the number of bikes at each station, but minimizes an objective that measures deviations from those targets. Similarly, Erdoğan et al. [2014] presumes a pre-determined range for how many bikes should be at
each station, while we only impose a lower bound and similarly to our model, Erdoğan et al. [2014] assumes rebalancing is done with a single vehicle that follows a fixed route. Finally, Dell’Amico et al. [2014] and Dell’Amico et al. [2016] propose models that minimize rebalancing costs with the use of a fleet of capacitated vehicles, instead of a unique vehicle.

2.2 Stochastic bikesharing problems

We next turn our attention to models that recognize uncertainty in how bikes will be used. 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 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. Brinkmann et al. [2015] present a stochastic model of the rebalancing problem in which they seek to minimize the expected number of starvations. They present heuristics for solving this model. Conversely, Datner et al. [2017] focuses solely on determining the initial allocation of bikes to stations, with the goal of minimizing the frequency of congestion and starvation events under demand realizations drawn from a stochastic process. However, in contrast to our model, they do not consider station capacities or that the fleet of bikes that may be allocated to stations can be of limited size.

Bike sharing systems are, in a sense, similar to car sharing systems. Fan [2014] propose a multistage stochastic model for allocating and redistributing cars wherein they consider uncertainty in the demand for cars at each “station”. Their model seeks to maximize expected profit, which is calculated as the difference between total revenue and rebalancing costs. However, they do not consider the quality of service provided by the system (i.e. the number of starvation or congestion events) and they use only three scenarios, while
our work presents an extensive study to better describe the uncertainty.

2.3 Dynamic rebalancing problem

In contrast to the static rebalancing problem, wherein it is assumed that rebalancing is done at night when no bikes are in use, researchers have also studied a dynamic rebalancing problem, wherein rebalancing is done while bikes are in use. As a result, in the dynamic problem, the locations of bikes may change during the rebalancing operation. For this problem, Ghosh et al. [2017] proposes a deterministic optimization model that seeks to maximize a measure of profit that is a function of both user ride length and rebalancing costs. Finally, Regue and Recker [2014], present a sequential framework for the dynamic rebalancing problem. 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 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.

3 Problem description

In this section, we provide a description of the problem. Specifically, we study a bikesharing 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 in the day, under uncertain demand. When making these decisions, the service provider seeks to both ensure that a customer who seeks to rent a bike from a specific station can do so, and, that a customer who seeks to return a bike to a specific station can do so. In addition, the service provider wishes to provide a high level of service with few bikes and little rebalancing. We consider a single depot (although the methods we propose could be extended to systems with
multiple depots) and multiple stations, which are represented by the set \( \mathcal{I} \). Since we presume rebalancing is done via a fixed route, \( \mathcal{I} \) is an ordered set, with the ordering determined by the sequence in which the route visits stations. Since the route ends at the depot, we consider the depot as such a station, and represent it by \( I \) (\( \in \mathcal{I} \)). We assume that there is a limited number of bikes \( I_{I0} \) at the depot that can be allocated to stations and that this number also corresponds to the depot capacity. On the other hand, each station \( i \in \mathcal{I} \setminus \{I\} \) is characterized by a bike capacity \( Q_i \) and by a number of bikes that is initially at that station, \( I_{i0} \) (which may be 0). We presume rebalancing is executed by a vehicle with capacity \( C \), that travels along a known and fixed route (determined \textit{a priori} by solving a Travelling Salesman Problem (TSP)) that begins at the depot, visits each station, and then ends at the depot. Finally, while each station has a capacity, it can also be expanded on a temporary basis using what has been referred to as a “valet service”, which involves stations being staffed with individuals who can accept returning bikes even when the station is at capacity.

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. At the beginning of the day (say 6 a.m.), the first stage decision is to determine the number of bikes to deliver from the depot to each station \( i \in \mathcal{I} \setminus \{I\} \) (before knowing the station demand) and we represent it with the variables \( x_i \). In order to ensure that a station can satisfy bike rental requests in the early hours, before any bikes are returned, we compute a minimum number of bikes that have to be allocated to each station, \( x_i \) (which may be 0).

We presume that, when the decision regarding the allocation of bikes to stations is made, only a probability distribution of station demand is known. Recalling that the bike demand level at each station \( i \in \mathcal{I} \setminus \{I\} \) is a random variable, we denote the set of
possible realizations (or scenarios) of the uncertain parameter with \( S \). Furthermore, we represent the stochastic bike demand at station \( i \in I \setminus \{I\} \) in scenario \( s \in S \) with \( d^s_i \) and the probability assigned to each scenario \( s \in S \) with \( pr^s \). Bike sharing systems are rental systems. Thus, product demand is different from what is considered in many inventory models wherein products are purchased, and from the supplier’s perspective, disappear. As such, we measure demand at the station level, and during the period between when bikes are initially allocated and redistributed, as the difference between the number of rented and returned bikes at that station during that period. Note, this means that \( d^s_i \) 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 \( s \)).

Then, we presume that at a later point in the day (say 12 p.m.), after the system has been in use and consequently stations’ demand has realized, the service provider observes the number of bikes at each station and determines a rebalancing plan. As a result, we model the determination of this rebalancing plan with second stage, scenario-dependent variables \( y^s_{i,i+1} \), which dictate the number of bikes to re-distribute from station \( i \in I \setminus \{I\} \) to the next station on the fixed route. We do not currently model the opportunity to allocate bikes from the depot to a station during the rebalancing operation. However, the model can be extended to accommodate that operation. We notice that doing so allows us to measure the ideal fleet size as the total number of bikes initially allocated to stations. On the other hand, if we allowed rebalancing from the depot, the number of allocated bikes could not be representative of the fleet size, as it could be adjusted according to scenario realization.

We illustrate the sequence of decisions and events in Figure 1.
We recall that our problem includes different 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). At the same time, the provider seeks to minimize both the number of bikes allocated to prevent bikes damage and rebalanced. To handle these different objectives, we next discuss how we measure each of these dimensions.

Starvation is measured by the number of bike rental requests at a station during the period of time between initial allocation and rebalancing that are in excess of the number of bikes that are positioned there at the end of the second stage, which we denote as $I_{s-}, i \in I \setminus \{I\}$. We refer to the weight associated with starvation as the “stock-out penalty” $p_i, i \in I \setminus \{I\}$. Congestion is measured with two terms, $B_i^{s+}, i \in I \setminus \{I\}$ and $E_i^{s+}, i \in I \setminus \{I\}$, with the first measuring the number of bikes at a station above and beyond the number initially allocated (which we term “extra inventory”), and the second measuring the number of bikes in excess of station capacity (which we term “excess...\text{Continued on next page.}
inventory”). We refer to the weight associated with “excess inventory” as the “excess penalty,” $c_i$, $\forall i \in I \setminus \{I\}$ and the weight associated with “extra inventory” as the “extra penalty,” $\frac{Q_i}{Q}$, $\forall i \in I \setminus \{I\}$.

We choose to model both extra and excess inventory for multiple reasons. First, we interpret the first stage decisions as indicating the inventory level for bikes at each station that best prevents both congestion and starvation. Thus, we penalize when there are more bikes (but fewer that the station capacity) at a station than in that initial allocation. Second, as we seek to limit the occurrence of congestion, we penalize if the surplus with respect to the station capacity occurs, since it implies higher operational costs due for having a staff member at a station (“valet service”).

Regarding the penalties associated with providing the service, the objective contains a term regarding initial allocation, which is measured as the number of bikes allocated to each station. A “delivery penalty”, $f_i$, $\forall i \in I \setminus \{I\}$, is associated with this number. Finally, the objective includes a term regarding re-balancing, which is measured by the total number of bikes moved between stations. The weight associated with this quantity is referred to as the “rebalancing penalty” $t_{i,i+1}$, $i \in I \setminus \{I\}$.

To address this problem, we now summarize the notation and we propose an integer non linear stochastic program, which we linearize (details in Appendix A) in order to solve it with an off-the-shelf mixed integer programming (MIP) program solver.

*Notation.*

*Sets:*

$I = \{1, \ldots, I\}$, set of stations (depot included);

$S = \{1, \ldots, S\}$, set of scenarios;

*Deterministic parameters:*

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

$\overline{T}_{I0}$, initial availability of bikes at the depot and depot capacity;
\( Q_i, \) capacity of station \( i \in I \setminus \{I\}; \\
\( I_{i0}, \) initial availability of bikes at station \( i \in I \setminus \{I\}; \\
C, \) capacity of the vehicle used for transshipment; \\
\( p_i, \) stock-out penalty at station \( i \in I \setminus \{I\}; \\
\( c_i, \) excess penalty at station \( i \in I \setminus \{I\}; \\
\( q_i, \) extra penalty at station \( i \in I \setminus \{I\}; \\
f_i, \) delivery penalty at station \( i \in I \setminus \{I\}; \\
t_{i,i+1}, \) rebalancing penalty at station \( i \in I \setminus \{I\}; \\

**Stochastic parameters:** \\
\( d_i^s, \) stochastic demand of bikes at station \( i \in I \setminus \{I\} \) in scenario \( s \in S; \\
p_r^s, \) probability of scenario \( s \in S; \\

**Variables:** \\
x_i \in \mathbb{Z}^+, \) first-stage variables representing the number of bikes to allocate at station \( i \in I \setminus \{I\}; \\
y_{i,i+1}^s \in \mathbb{Z}^+, \) second-stage variables representing the number of bikes to relocate from station \( i \in I \setminus \{I\} \) to station \( i + 1; \\
I_i^s \in \mathbb{Z}, \) balance of bikes at station \( i \in I \setminus \{I\} \) in scenario \( s \in S; \\
I_i^{s+} \in \mathbb{Z}^+, \) units of surplus at station \( i \in I \setminus \{I\} \) in scenario \( s \in S; \\
I_i^{s-} \in \mathbb{Z}^-, \) units of stock-out at station \( i \in I \setminus \{I\} \) in scenario \( s \in S; \\
B_i^s \in \mathbb{Z}, \) extra inventory balance at station \( i \in I \setminus \{I\} \) in scenario \( s \in S; \\
B_i^{s+} \in \mathbb{Z}^+, \) units of extra inventory at station \( i \in I \setminus \{I\} \) in scenario \( s \in S; \\
E_i^s \in \mathbb{Z}, \) excess inventory balance at station \( i \in I \setminus \{I\} \) in scenario \( s \in S; \\
E_i^{s+} \in \mathbb{Z}^+, \) units of excess inventory at station \( i \in I \setminus \{I\} \) in scenario \( s \in S; \\

The model is:
Problem B

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

s.t:

\[ x_i \geq x_i^0 \quad i \in I \backslash \{I\} \]
\[ \bar{T}_{i0} + x_i \leq Q_i \quad i \in I \backslash \{I\} \]
\[ \sum_{i \in I \backslash \{I\}} x_i \leq \bar{T}_{i0} \]
\[ y_{i,i+1}^s \leq C \quad i \in I \backslash \{I\}, \quad s \in S \]
\[ I_i^s = \bar{T}_{i0} - \sum_{i \in I \backslash \{I\}} x_i + y_{i-1,I}^s \quad s \in S \]
\[ I_i^s \leq \bar{T}_{i0} \quad s \in S \]
\[ I_i^s = \bar{T}_{i0} + x_1 - d_1^s - y_{1,2}^s \quad s \in S \]
\[ I_i^s = \bar{T}_{i0} + x_i - d_i^s + y_{i-1,i}^s - y_{i,i+1}^s \quad i \in I \backslash \{1, I\}, \quad s \in S \]
\[ I_i^{s+} = \max\{0, I_i^s\} \quad i \in I \backslash \{I\}, \quad s \in S \]
\[ I_i^{s-} = \min\{0, I_i^s\} \quad i \in I \backslash \{I\}, \quad s \in S \]
\[ E_i^s = I_i^{s+} - Q_i \quad i \in I \backslash \{I\}, \quad s \in S \]
\[ E_i^{s+} = \max\{0, E_i^s\} \quad i \in I \backslash \{I\}, \quad s \in S \]
\[ B_i^s = I_i^{s+} - x_i - \bar{T}_{i0} - E_i^{s+} \quad i \in I \backslash \{I\}, \quad s \in S \]
\[ B_i^{s+} = \max\{0, B_i^s\} \quad i \in I \backslash \{I\}, \quad s \in S \]
\[ x_i \geq 0 \quad \text{integer} \quad i \in I \backslash \{I\} \]
\[ y_{i,i+1}^s \geq 0 \quad \text{integer} \quad i \in I \backslash \{I\}, \quad s \in S \]
\[ I_i^s \quad \text{free and integer} \quad i \in I \backslash \{I\}, \quad s \in S \]
\[ I_i^{s^+} \geq 0 \text{ integer} \quad i \in I \setminus \{I\}, \ s \in \mathcal{S} \]  
(19)

\[ I_i^{s^-} \leq 0 \text{ integer} \quad i \in I \setminus \{I\}, \ s \in \mathcal{S} \]  
(20)

\[ B_i^s \text{ free and integer} \quad i \in I \setminus \{I\}, \ s \in \mathcal{S} \]  
(21)

\[ B_i^{s^+} \geq 0 \text{ integer} \quad i \in I \setminus \{I\}, \ s \in \mathcal{S} \]  
(22)

\[ E_i^s \text{ free and integer} \quad i \in I \setminus \{I\}, \ s \in \mathcal{S} \]  
(23)

\[ E_i^{s^+} \geq 0 \text{ integer} \quad i \in I \setminus \{I\}, \ s \in \mathcal{S} \]  
(24)

The objective function (1) represents the minimization of the total expected penalty, obtained through the sum of all the charged penalties for delivery, rebalancing, extra and excess inventory and stock-out. 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 $s \in S$. 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 $s \in S$. Constraints (10) and (11) determine the surplus and stock-out quantities, respectively, for each station and for each scenario $s \in S$. Recalling that we model the presence of a valet service, wherein bikes can be returned to stations that are full, constraints (12) and (13) calculate the number of bikes at each station that are in excess of station capacity, in each scenario $s \in S$. Similarly, constraints (14) and (15) determine, for each scenario $s \in S$, when there are more bikes positioned at a station after rebalancing than were initially allocated, but not more than station capacity. If there is, these constraints ensure that $B_i^\text{surplus}$ represents that number of bikes. Finally, Constraints (16) to (24) are variable definition constraints.

We notice that, in this model, infeasibility cannot occur since there are no constraints imposing the demand satisfaction and if there are no bikes to satisfy the demand or if the station is more than full, a penalty is added in the objective function.

### 5 Numerical Results

In this section, we present and analyze the results of our computational study. With this study, we seek to assess the benefits of modeling uncertainty. We also compare the quality of the initial allocation plan prescribed by the solution to the stochastic program with an estimate based on historical data of how bikes were allocated in practice.

Our computational study is based on the bike sharing system in San Francisco, CA. Specifically, we use publicly available data (open data) to simulate the use of different initial allocation plans for that system, and calculate performance metrics related to customer service and cost. As such, we first illustrate the state of art of the San Francisco
system which also represents the motivation for our work. After that, we describe how we generated an instance of the stochastic program based upon this system, as well as how we simulated the use of those initial allocation plans in the context of that system. We then present the results of our analysis. 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 with Gurobi 7.5.1.

5.1 State of art of San Francisco bikesharing system

To estimate the magnitude of congestion and starvation in a realistic setting, we simulated the bike sharing system of the city of San Francisco. This simulation was based on publicly available data regarding ridership and bike availabilities in the time interval of 6 am to 11:59 am for the months of May through August, 2016. We illustrate in Figure 2a the percentage of minutes with congestion (computed with respect to the overall considered time interval), by station, and with separate graphs for weekdays and weekends. We see that congestion is more likely during the week, with some stations seeing an event 20% of the time. Figure 2b is similar, only it reports the percentage of minutes with starvation. Here we see an even greater frequency, with some stations experiencing a starvation event in over 30% of the simulated overall time interval.
Figure 2: Congestion and starvation average frequencies in the real case for year 2016, considering the months from May to August and the time interval between 6 a.m. and 11.59 a.m.

5.2 Test setting

Figure 3 illustrates the bike sharing system of the city of San Francisco, with markers indicating its 33 stations (in August, 2016). Ridership data for this system can be found at the website www.bayareabikeshare.com/open-data. We derived our instances, and simulation model, from ridership data from the summer months (May through August) in the period of time between August, 2013 (the launch of the bike sharing service) and August 2016. We note that when we first accessed the website, it was indicated that the system consisted of 350 bikes, and we used that number in both settings (instances and simulation model).
In our computational study, both the allocation penalty, $f_i$, and rebalancing penalty, $t_{i,i+1}$ are set to the same value for each station. Specifically, we set $f_i = 1$, and $t_{i,i+1} = 2$ for all stations $i$. 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. As such, we set both $c_i = p_i = \kappa(1 + \min_{j \in \mathcal{I}, j \neq i} \delta_{ij})$, for all stations $i$, where $\delta_{ij}$ is the distance between station $i$ and $j$ calculated as the geodesic distance using the great-circle distance formula (Banerjee [2005]). Specifically, we set $\kappa = 46$, so that, in most of cases, the model redistributes bikes from a station when there is extra inventory to one of the following stations (since it is cheaper to rebalance than having congestion or starvation). Regarding other model parameter values, we presume the vehicle used for rebalancing bikes has a capacity of 25 bikes. This is the same value that was used in Forma et al. [2015].

5.2.1 Scenario generation

Scenario generation is an important part of the modelling process, since a bad scenario tree can lead to a not meaningful solution of the optimization problem. 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 process demand $d^n_i, i \in I, s \in S$, 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’s 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 $d^n_i, i \in I \setminus \{I\}, n = 1, \ldots, N$, 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. Results show that the two patterns follow a similar behaviour with only a difference in scale, due to the number of considered samples.
Figure 4: Empirical distributions and Monte Carlo sampling for two different stations
Pseudo-code 1: Scenario generation process

1 Input: $d^n_i \in \mathbb{Z}, i \in I \setminus \{I\}, n = 1, \ldots, N$

2 for $i \in I \setminus \{I\}$ do

3 $\mathcal{K} := \{n', n'' = 1, \ldots, N : d^n_{i'} \neq d^n_{i''}\}$

4 if $d^n_{i'} < d^n_{i''}$ then

5 $n' < n''$

6 end

7 for $k \in \mathcal{K}$ do

8 $cdf_i[k] := \frac{\sum_{n', n'' = 1, \ldots, N} \mathbf{1}(d^n_{i'} = d^n_{i''})}{N}$

9 $inv.cdf_i[k] := d^k_i$

10 end

11 for $s \in \mathcal{S}$ do

12 sample a random number ("random") in $[0,1]$

13 for $k \in \mathcal{K}$ do

14 if $random <= cdf_i[k]$ and $random > cdf_i[k - 1]$ then

15 $d^s_i = inv.cdf_i[k]$

16 end

17 end

18 return $d^s_i$

19 end

20 end

5.2.2 Determining the size of the scenario tree

In order to understand how many scenarios are needed to obtain a stable objective function, 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 is nearly
monotonic and decreasing with a percentage gap under 0.1%. We notice that, even if the
model is an integer linear program, an optimal integer solution has always been obtained
by solving the continuous relaxation at the root node. As a consequence, considering that
the continuous relaxation of the model with 1,200 scenarios required 66 seconds to solve,
whereas one with 2,000 required 115 seconds, we base our computational study on a set
of 1,200 scenarios. We note that we presume the scenarios are equi-probable, i.e. we set
\[ p_r^s = \frac{1}{|\mathcal{S}|}, \quad \forall s \in \mathcal{S}. \]

Notice that, 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 what could be thought of as a “steady-state” demand and ignores the dynamics of the system. As an example, consider a station where first a bike is withdrawn and then one is returned, and no other bikes are withdrawn or returned. We would compute the resulting station demand as 0, which in turn would indicate to the model that no bikes need to be allocated to that station (unless they were allocated to that station only to be rebalanced later). With zero bikes allocated, however, the first withdrawal request can not be satisfied. As such, we discuss in Appendix B our method for estimating the number of bikes withdrawn from a station before any are returned, from which we derive the parameter values $x_i$.

5.2.3 Simulation of the inventory levels

We also use historical ridership data to simulate the performance of our initial allocation of bikes to stations during the week of June 20, 2016 to June 26, 2016. Given a day during this week, we simulate the movement of bikes based on rides taken on that day, and record statistics related to congestion and starvation. Specifically, regarding congestion, we record the number of times during the simulation a user wants to return a bike to a station, but it is full. Similarly, regarding starvation, we record the number of times during the simulation a user wants to withdraw a bike from a station, but it is empty. Then, we have a final inventory level at each station, $I_{i}^{final} = I_{i0} + x_i - w_i + s_i$, 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 which we solve the second stage of our stochastic program in order to determine how rebalancing should be done and the number of bikes rebalanced. Similarly, we calculate the total number of rebalanced bike miles, 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.
Since it could be interesting to observe how the system performs after rebalancing is done, 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 they can satisfy. We base this estimate on the same set of scenarios described above. For a given station, \( i \), and scenario \( s \), we compute the fill rate as follows:

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

(25)

We then compute an overall fill rate by averaging this statistic over all stations and scenarios.

5.3 Analyzing the value of uncertainty and the quality of the expected value solution

To assess the value of modeling uncertainty, we compute the Value of the Stochastic Solution (VSS) (Birge and Louveaux [2011], Kall and Wallace [1994], Maggioni et al. [2016]). 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 variant of our stochastic program, in which the random demand parameters are replaced with their expected values, rounded to the nearest integer (bikes demands cannot be represented 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 our simulation model, and report statistics related to the performance of each plan in Table 1. We see that while the stochastic program allocates 40% more bikes, it leads to a much smaller frequency of starvation and higher fill rate. 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.

Table 1: Simulation-based comparison of solutions to stochastic and deterministic problems.

<table>
<thead>
<tr>
<th></th>
<th>SP</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>% Congestion</td>
<td>% Starvation</td>
<td>Miles Rebalanced Qty</td>
<td>Expected Fill Rate</td>
<td>Extra Inventory Qty</td>
<td>% Congestion</td>
<td>% Starvation</td>
<td>Miles Rebalanced Qty</td>
<td>Expected Fill Rate</td>
<td>Extra Inventory Qty</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mon</td>
<td>8.81%</td>
<td>31.30%</td>
<td>19.95</td>
<td>85.86%</td>
<td>149</td>
<td>5.56%</td>
<td>39.57%</td>
<td>56.65</td>
<td>86.28%</td>
<td>151</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tue</td>
<td>9.21%</td>
<td>28.74%</td>
<td>22.27</td>
<td>86.28%</td>
<td>149</td>
<td>6.53%</td>
<td>38.32%</td>
<td>49.70</td>
<td>86.15%</td>
<td>153</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wed</td>
<td>7.27%</td>
<td>31.11%</td>
<td>16.51</td>
<td>87.55%</td>
<td>158</td>
<td>4.44%</td>
<td>40.50%</td>
<td>46.55</td>
<td>88.13%</td>
<td>163</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thu</td>
<td>6.94%</td>
<td>27.88%</td>
<td>15.36</td>
<td>86.43%</td>
<td>141</td>
<td>4.49%</td>
<td>36.01%</td>
<td>42.74</td>
<td>86.17%</td>
<td>145</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fri</td>
<td>6.68%</td>
<td>25.98%</td>
<td>19.81</td>
<td>86.42%</td>
<td>105</td>
<td>3.23%</td>
<td>36.55%</td>
<td>51.50</td>
<td>87.03%</td>
<td>109</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sat</td>
<td>7.62%</td>
<td>21.36%</td>
<td>3.93</td>
<td>88.22%</td>
<td>36</td>
<td>0.00%</td>
<td>23.30%</td>
<td>3.36</td>
<td>85.26%</td>
<td>37</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sun</td>
<td>0.00%</td>
<td>20.83%</td>
<td>2.37</td>
<td>91.32%</td>
<td>19</td>
<td>0.00%</td>
<td>25%</td>
<td>2.31</td>
<td>88.04%</td>
<td>19</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>6.65%</td>
<td>26.74%</td>
<td>14.27</td>
<td>87.44%</td>
<td>108.29</td>
<td>3.46%</td>
<td>34.18%</td>
<td>35.28</td>
<td>86.77%</td>
<td>111</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

We next seek to understand why the solution to the EV problem performs poorly in comparison to the solution to the stochastic program. 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].
To compute the $LUSS$, we examine the solution to the $EV$ problem to determine the subset of stations $\bar{I}$ to which it allocates more bikes than their initial requirement $x_i$. We then solve the stochastic program, fixing $x_i = \bar{x}_i$ for stations $i \in I \setminus \bar{I}$. 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 $\bar{I}$ 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 can 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 3 stations receive a higher number of bikes than the initial requirement, compared to the $SP$ in which these stations are 12. We also note that the 3 stations activated in the $EV$ solution are the same activated in the $SP$ solution.
This result could justify a deeper investigation of the quality of the Expected Value Solution by computing the \textit{Loss of Reduced Costs-based Variable Fixing (LRCVF)} as the difference between the optimal values of the stochastic problem and its reduced version, obtained by fixing a certain number of variables taking into account the information from the reduced cost of the expected value solution (for more details see Crainic et al. [2018]).

With the \textit{LUDS}, we seek to determine whether the solution to the \textit{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 \textit{EV} problem. We refer to the objective function value of the optimal solution to this restricted stochastic program as the \textit{Expected Input Value (EIV)} and compute a relative \textit{LUDS} measure as

\[
LUDS = \frac{(EIV - RP)}{RP} = 0\%.
\]
The result means that the EV solution is perfectly upgradable, 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.

Finally, we study the correlation between the variance in demand at a station and the number of bikes allocated to that station in both the solution to the stochastic program and the solution to the EV problem. We present a scatter plot of demand variance against number of allocated bikes in Figure 7. We observe a high correlation (0.84) in the solution to the stochastic program, suggesting that the stochastic program allocates more bikes to stations that have a greater variability in station demand. However, we also see a correlation of 0.64 in the solution to the EV problem. While this may seem counter-intuitive, as the EV problem does not recognize variance, there is a high correlation between the average and variance of demand at stations. Thus, we conclude that this correlation coefficient of 0.64 reflects the solution to the EV problem allocating bikes to stations with high average demand.
5.4 A comparison with the implemented system

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 initial 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 two initial allocation plans with our simulation model. We present statistics regarding how each plan performed in Table 2. We see that the stochastic program allocated far fewer bikes (45% fewer), which in turn lead to less congestion and rebalancing. However, not surprisingly, we saw an increase in the starvation frequency.

Next, to normalize our comparison, we add a constraint to the stochastic program to ensure that it allocates the same total number of bikes on each day as used in the real system. We present the statistics related to that plan in Table 3. Here, we see that the allocation plans prescribed by the stochastic program outperform the actual allocation on each day and in each category. We view these results as a strong indicator of the impact
Table 2: Comparison of implemented plan and plan from stochastic program

<table>
<thead>
<tr>
<th></th>
<th>Total bikes at 6 a.m.</th>
<th>Congestion</th>
<th>Starvation</th>
<th>Miles * Rebalanced Qty</th>
<th>Expected Fill Rate</th>
<th>Extra Inventory Qty</th>
<th>Total bikes at 6 a.m.</th>
<th>Congestion</th>
<th>Starvation</th>
<th>Miles * Rebalanced Qty</th>
<th>Expected Fill Rate</th>
<th>Extra Inventory Qty</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mon</td>
<td>272</td>
<td>17.24%</td>
<td>25.39%</td>
<td>40.89</td>
<td>90.92%</td>
<td>113</td>
<td>155</td>
<td>8.81%</td>
<td>31.30%</td>
<td>95.86%</td>
<td>22.17</td>
<td>6.28%</td>
</tr>
<tr>
<td>Tue</td>
<td>275</td>
<td>14.97%</td>
<td>19.76%</td>
<td>38.19</td>
<td>88.58%</td>
<td>109</td>
<td>155</td>
<td>9.21%</td>
<td>28.74%</td>
<td>96.28%</td>
<td>22.17</td>
<td>6.28%</td>
</tr>
<tr>
<td>Wed</td>
<td>281</td>
<td>15.76%</td>
<td>23.36%</td>
<td>24.40</td>
<td>91.48%</td>
<td>118</td>
<td>155</td>
<td>7.47%</td>
<td>31.11%</td>
<td>97.55%</td>
<td>16.85</td>
<td>8.57%</td>
</tr>
<tr>
<td>Thu</td>
<td>289</td>
<td>13.88%</td>
<td>22.22%</td>
<td>38.76</td>
<td>90.92%</td>
<td>101</td>
<td>155</td>
<td>6.94%</td>
<td>27.88%</td>
<td>96.43%</td>
<td>15.34</td>
<td>8.57%</td>
</tr>
<tr>
<td>Fri</td>
<td>285</td>
<td>10.14%</td>
<td>14.94%</td>
<td>22.73</td>
<td>91.46%</td>
<td>85</td>
<td>155</td>
<td>6.68%</td>
<td>25.98%</td>
<td>96.42%</td>
<td>19.89</td>
<td>8.57%</td>
</tr>
<tr>
<td>Sat</td>
<td>293</td>
<td>4.76%</td>
<td>2.91%</td>
<td>3.36</td>
<td>94.03%</td>
<td>31</td>
<td>155</td>
<td>7.62%</td>
<td>21.36%</td>
<td>88.26%</td>
<td>3.36</td>
<td>8.57%</td>
</tr>
<tr>
<td>Sun</td>
<td>288</td>
<td>17.39%</td>
<td>14.58%</td>
<td>2.12</td>
<td>92.80%</td>
<td>18</td>
<td>155</td>
<td>0.00%</td>
<td>20.83%</td>
<td>91.34%</td>
<td>2.31</td>
<td>8.57%</td>
</tr>
<tr>
<td>Average</td>
<td>283.29</td>
<td>13.45%</td>
<td>17.58%</td>
<td>24.24</td>
<td>91.46%</td>
<td>82.14</td>
<td>155</td>
<td>6.68%</td>
<td>26.74%</td>
<td>85.45%</td>
<td>14.27</td>
<td>8.57%</td>
</tr>
</tbody>
</table>

Table 3: Comparison of implemented plan and plan from stochastic program, when allocating same number of bikes.

<table>
<thead>
<tr>
<th></th>
<th>Total bikes at 6 a.m.</th>
<th>Congestion</th>
<th>Starvation</th>
<th>Miles * Rebalanced Qty</th>
<th>Expected Fill Rate</th>
<th>Extra Inventory Qty</th>
<th>Total bikes at 6 a.m.</th>
<th>Congestion</th>
<th>Starvation</th>
<th>Miles * Rebalanced Qty</th>
<th>Expected Fill Rate</th>
<th>Extra Inventory Qty</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mon</td>
<td>272</td>
<td>17.24%</td>
<td>25.39%</td>
<td>40.89</td>
<td>90.92%</td>
<td>113</td>
<td>272</td>
<td>12.07%</td>
<td>18.70%</td>
<td>28.21</td>
<td>91.92%</td>
<td>127</td>
</tr>
<tr>
<td>Tue</td>
<td>275</td>
<td>14.97%</td>
<td>19.76%</td>
<td>38.19</td>
<td>88.58%</td>
<td>109</td>
<td>275</td>
<td>11.90%</td>
<td>18.56%</td>
<td>28.21</td>
<td>91.92%</td>
<td>117</td>
</tr>
<tr>
<td>Wed</td>
<td>281</td>
<td>15.76%</td>
<td>23.36%</td>
<td>24.40</td>
<td>91.48%</td>
<td>118</td>
<td>281</td>
<td>9.70%</td>
<td>22.97%</td>
<td>28.21</td>
<td>91.92%</td>
<td>117</td>
</tr>
<tr>
<td>Thu</td>
<td>289</td>
<td>13.88%</td>
<td>22.22%</td>
<td>38.76</td>
<td>90.92%</td>
<td>101</td>
<td>289</td>
<td>9.18%</td>
<td>16.99%</td>
<td>28.21</td>
<td>91.92%</td>
<td>117</td>
</tr>
<tr>
<td>Fri</td>
<td>285</td>
<td>10.14%</td>
<td>14.94%</td>
<td>22.73</td>
<td>91.46%</td>
<td>85</td>
<td>285</td>
<td>7.83%</td>
<td>13.79%</td>
<td>28.21</td>
<td>91.92%</td>
<td>117</td>
</tr>
<tr>
<td>Sat</td>
<td>293</td>
<td>4.76%</td>
<td>2.91%</td>
<td>3.36</td>
<td>94.03%</td>
<td>31</td>
<td>293</td>
<td>7.62%</td>
<td>21.36%</td>
<td>28.21</td>
<td>91.92%</td>
<td>117</td>
</tr>
<tr>
<td>Sun</td>
<td>288</td>
<td>17.39%</td>
<td>14.58%</td>
<td>2.12</td>
<td>92.80%</td>
<td>18</td>
<td>288</td>
<td>4.35%</td>
<td>8.33%</td>
<td>28.21</td>
<td>91.92%</td>
<td>117</td>
</tr>
<tr>
<td>Average</td>
<td>283.29</td>
<td>13.45%</td>
<td>17.58%</td>
<td>24.24</td>
<td>91.46%</td>
<td>82.14</td>
<td>155</td>
<td>6.68%</td>
<td>26.74%</td>
<td>85.45%</td>
<td>14.27</td>
<td>81.86%</td>
</tr>
</tbody>
</table>

6 Managerial insights

In this section, we present some managerial insights which can be valuable for practitioners.

From the literature and from the case study of the bikesharing system of San Francisco.
(see subsection 5.1), one of the problems in common to all bikesharing systems is that stations are full or empty very often. For this reason, we believe that our approach could be extended to other bikesharing systems and, without loss of generality, the following managerial insights could be valid.

First, managers should apply a stochastic model since demand is uncertain. By applying a deterministic model, the solution is very sub-optimal, since it delivers too few bikes to too few stations, leading to a higher frequency of starvation, more rebalancing and inventory and a lower demand fill rate. However, practitioners could reduce the computational effort to obtain the solution of the stochastic program by upgrading the deterministic solution. The heuristic procedure we proposed can be valuable, especially if this problem must be solved multiple times in a day. Second, the number of allocated bikes represents an indicator of how big the fleet size should be. According to this, managers should carefully evaluate whether to allocate the total number of available bikes at the depot. In our case study, only a fraction of the total availability of bikes should be delivered in order to reduce the rebalancing and the risks of congestion and those deriving from the allocation.

Third, rebalancing is fundamental in order to adjust the allocation decision over time. As a matter of fact, this strategy is able to introduce more flexibility to better manage a bikesharing system. Finally, managers should carefully select the stations towards which bikes should be delivered, as each station is characterized by different features, such as the number of requests for bikes and free docks, capacity and distance to the closest station.

In our work, we captured all these elements and through the comparison of the allocation decisions suggested by our stochastic program and those actually implemented, we observed that our solution improves the allocation of bikes between stations, as it leads to lower frequencies of congestion and starvation, lower rebalancing and a higher expected fill rate.
7 Conclusions and future works

In this paper, we studied the problem of determining an initial allocation of bikes to stations, as well as the opportunity to perform re-balancing 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. 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. Finally, we proposed managerial insights which could be valuable for practitioners in order to manage a general bikesharing system.

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 will 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 the possibility 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


Teodor Gabriel Crainic, Francesca Maggioni, Guido Perboli, and Walter Rei. Reduced

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


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:

\[
\min \sum_{i \in I} f_i x_i + \sum_{s \in S} p_r [ \sum_{i \in I \setminus \{I\}} (t_{i,i+1} + y_{i,i+1} + \frac{c_i}{Q_i} B_{i}^{s+} + c_i E_{i}^{s+} + p_i I_{i}^{s-})].
\]  

(26)

Then, we need to linearize the expressions for determining \(I_i^s\) and \(I_i^{s-}\). We introduce the binary variable \(z_i^s\) such that:

\[
z_i^s = \begin{cases} 
1 & \text{if } I_i^s \geq 0 \\
0 & \text{otherwise}
\end{cases}
\]

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

\[
I_i^s \geq 0 \text{ integer } 
i \in I \setminus \{I\}, s \in S \]  

(27)

\[
I_i^s \leq I_i^s + \max_{s \in S} (d_i^s)(1 - z_i^s) \ni \in I \setminus \{I\}, s \in S
\]  

(28)

\[
I_i^s \geq 0 \text{ integer } 
i \in I \setminus \{I\}, s \in S
\]  

(29)

\[
I_i^s \geq M z_i^s \ni \in I \setminus \{I\}, s \in S
\]  

(30)

\[
I_i^s \geq -I_i^s \ni \in I \setminus \{I\}, s \in S
\]  

(31)

\[
I_i^s \leq -I_i^s + M z_i^s \ni \in I \setminus \{I\}, s \in S
\]  

(32)

\[
I_i^s \geq 0 \text{ integer } 
i \in I \setminus \{I\}, s \in S
\]  

(33)

\[
I_i^s \leq \max_{s \in S} (d_i^s)(1 - z_i^s) \ni \in I \setminus \{I\}, s \in S
\]  

(34)

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:

\[
e_i^s = \begin{cases} 
1 & \text{if } E_i^s \geq 0 \\
0 & \text{otherwise}
\end{cases}
\]

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

\[
E_i^s \geq 0 \text{ integer } 
i \in I \setminus \{I\}, s \in S
\]  

(35)

\[
E_i^s \leq E_i^s + \max_{s \in S} (d_i^s)(1 - z_i^s) \ni \in I \setminus \{I\}, s \in S
\]  

(36)

\[
E_i^s \geq M z_i^s \ni \in I \setminus \{I\}, s \in S
\]  

(37)

\[
E_i^s \geq -E_i^s \ni \in I \setminus \{I\}, s \in S
\]  

(38)

\[
E_i^s \leq -E_i^s + M z_i^s \ni \in I \setminus \{I\}, s \in S
\]  

(39)

\[
E_i^s \geq 0 \text{ integer } 
i \in I \setminus \{I\}, s \in S
\]  

(40)

\[
E_i^s \leq \max_{s \in S} (d_i^s)(1 - z_i^s) \ni \in I \setminus \{I\}, s \in S
\]  

(41)

The same linearization technique is applied for \(B_i^s\). In particular, we introduce the binary variables \(e_i^s\) and \(r_i^s\), such that:

\[
e_i^s = \begin{cases} 
1 & \text{if } E_i^s \geq 0 \\
0 & \text{otherwise}
\end{cases}
\]

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

\[
B_i^s \geq 0 \text{ integer } 
i \in I \setminus \{I\}, s \in S
\]  

(42)

\[
B_i^s \leq B_i^s + \max_{s \in S} (d_i^s)(1 - z_i^s) \ni \in I \setminus \{I\}, s \in S
\]  

(43)

\[
B_i^s \geq M z_i^s \ni \in I \setminus \{I\}, s \in S
\]  

(44)

\[
B_i^s \geq -B_i^s \ni \in I \setminus \{I\}, s \in S
\]  

(45)

\[
B_i^s \leq -B_i^s + M z_i^s \ni \in I \setminus \{I\}, s \in S
\]  

(46)

\[
B_i^s \geq 0 \text{ integer } 
i \in I \setminus \{I\}, s \in S
\]  

(47)

\[
B_i^s \leq \max_{s \in S} (d_i^s)(1 - z_i^s) \ni \in I \setminus \{I\}, s \in S
\]  

(48)
\[ r_i^s = \begin{cases} 1 & \text{if } B_i^s \geq 0 \\ 0 & \text{otherwise} \end{cases} \]

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

\[ B_i^{s+} \geq B_i^s \quad i \in \mathcal{I} \setminus \{I\}, \ s \in \mathcal{S} \] (35)

\[ B_i^{s+} \leq B_i^s + M(1 - r_i^s) \quad i \in \mathcal{I} \setminus \{I\}, \ s \in \mathcal{S} \] (36)

\[ B_i^{s+} \leq Mr_i^s \quad i \in \mathcal{I} \setminus \{I\}, \ s \in \mathcal{S} \] (37)

\[ B_i^{s+} \geq 0 \quad \text{integer} \quad i \in \mathcal{I} \setminus \{I\}, \ s \in \mathcal{S} \] (38)

\[ B_i^{s-} \geq -B_i^s \quad i \in \mathcal{I} \setminus \{I\}, \ s \in \mathcal{S} \] (39)

\[ B_i^{s-} \leq -B_i^s + Mr_i^s \quad i \in \mathcal{I} \setminus \{I\}, \ s \in \mathcal{S} \] (40)

\[ B_i^{s-} \leq M(1 - r_i^s) \quad i \in \mathcal{I} \setminus \{I\}, \ s \in \mathcal{S} \] (41)

\[ B_i^{s-} \geq 0 \quad \text{integer} \quad i \in \mathcal{I} \setminus \{I\}, \ s \in \mathcal{S} \] (42)

\[ E_i^{s+} \geq E_i^s \quad i \in \mathcal{I} \setminus \{I\}, \ s \in \mathcal{S} \] (43)

\[ E_i^{s+} \leq E_i^s + M(1 - e_i^s) \quad i \in \mathcal{I} \setminus \{I\}, \ s \in \mathcal{S} \] (44)

\[ E_i^{s+} \leq Me_i^s \quad i \in \mathcal{I} \setminus \{I\}, \ s \in \mathcal{S} \] (45)

\[ E_i^{s+} \geq 0 \quad \text{integer} \quad i \in \mathcal{I} \setminus \{I\}, \ s \in \mathcal{S} \] (46)

\[ E_i^{s-} \geq -E_i^s \quad i \in \mathcal{I} \setminus \{I\}, \ s \in \mathcal{S} \] (47)

\[ E_i^{s-} \leq -E_i^s + Me_i^s \quad i \in \mathcal{I} \setminus \{I\}, \ s \in \mathcal{S} \] (48)

\[ E_i^{s-} \leq M(1 - e_i^s) \quad i \in \mathcal{I} \setminus \{I\}, \ s \in \mathcal{S} \] (49)

\[ E_i^{s-} \geq 0 \quad \text{integer} \quad i \in \mathcal{I} \setminus \{I\}, \ s \in \mathcal{S} \] (50)
Finally, we introduce the variable definition constraints:

\[ z_i^s \in \{0, 1\} \quad i \in \mathcal{I} \setminus \{I\}, \; s \in \mathcal{S} \]  \hspace{1cm} (51)

\[ r_i^s \in \{0, 1\} \quad i \in \mathcal{I} \setminus \{I\}, \; s \in \mathcal{S} \]  \hspace{1cm} (52)

\[ e_i^s \in \{0, 1\} \quad i \in \mathcal{I} \setminus \{I\}, \; s \in \mathcal{S} \]  \hspace{1cm} (53)

**Appendix B  Determining the initial bike requirement for each 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 the 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 4. 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 4: Congestion and starvation relative frequencies by method of determining initial bike requirement

<table>
<thead>
<tr>
<th></th>
<th>Method 1</th>
<th>Method 2</th>
<th>Method 3</th>
<th>Method 4</th>
<th>Method 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>% Congestion</td>
<td>% Starvation</td>
<td>% Congestion</td>
<td>% Starvation</td>
<td>% Congestion</td>
<td>% Starvation</td>
</tr>
<tr>
<td>Mon</td>
<td>8.43%</td>
<td>33.07%</td>
<td>9.00%</td>
<td>29.53%</td>
<td>8.43%</td>
</tr>
<tr>
<td>Tue</td>
<td>8.64%</td>
<td>30.94%</td>
<td>9.41%</td>
<td>28.74%</td>
<td>8.64%</td>
</tr>
<tr>
<td>Wed</td>
<td>7.07%</td>
<td>32.57%</td>
<td>7.07%</td>
<td>30.48%</td>
<td>7.07%</td>
</tr>
<tr>
<td>Thu</td>
<td>6.94%</td>
<td>30.19%</td>
<td>7.14%</td>
<td>28.36%</td>
<td>6.94%</td>
</tr>
<tr>
<td>Fri</td>
<td>6.68%</td>
<td>28.97%</td>
<td>6.68%</td>
<td>28.97%</td>
<td>6.68%</td>
</tr>
<tr>
<td>Sat</td>
<td>7.62%</td>
<td>32.84%</td>
<td>7.62%</td>
<td>31.61%</td>
<td>7.62%</td>
</tr>
<tr>
<td>Sun</td>
<td>0.00%</td>
<td>29.17%</td>
<td>0.00%</td>
<td>20.83%</td>
<td>0.00%</td>
</tr>
<tr>
<td>Average</td>
<td>6.48%</td>
<td>30.99%</td>
<td>6.48%</td>
<td>31.46%</td>
<td>6.48%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Method 1</th>
<th>Method 2</th>
<th>Method 3</th>
<th>Method 4</th>
<th>Method 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial requirement</td>
<td>55</td>
<td>36</td>
<td>103</td>
<td>57</td>
<td>89</td>
</tr>
<tr>
<td>Total delivery</td>
<td>147</td>
<td>146</td>
<td>160</td>
<td>148</td>
<td>154</td>
</tr>
</tbody>
</table>