Submitted to *Transportation Science*
manuscript (Please, provide the mansucript number!)

Setting Inventory Levels in a Bike Sharing Network

Submitted: December 2015, Revised: March 2017, June 2017

Accepted: June 2017

Sharon Datner, Tal Raviv, Michal Tzur
Industrial Engineering Department, Tel Aviv University, 69978 Tel Aviv, Israel
sharondatner@gmail.com, talraviv@eng.tau.ac.il, tzur@eng.tau.ac.il

Daniel Chemla
Smart-Flows Ltd; Université Paris Est, CERMICS; Université Paris 13, Paris, France
dchemla@smart-flows.com

Bike Sharing Systems (BSSs) allow customers to rent bicycles at automatic rental stations distributed throughout a city, use them for a short period of time, and return them to any station. One of the major issues that BSS operators must address is non-homogeneous asymmetric demand processes. These demand processes create an inherent imbalance, thus leading to shortages either of bicycles when users are attempting to rent them and of vacant lockers when users are attempting to return them. The predominant approach taken by operators to cope with this difficulty is to reposition bicycles to rebalance the inventory levels at the different stations. Most repositioning studies assume that a target inventory level or range of inventory levels is known for each station. In this paper, we focus on determining the correct target level for repositioning according to a well-defined objective. This is a challenging task because of the nature of the user behavior that creates the interactions among the inventory levels at different stations. For example, if bicycles are not available at the user's origin, the user may either abandon the system, use other means of transportation, or look for available bicycles at a neighboring station. If in another case, a locker is not available at a user's destination, then that user is obliged to find a station with available space to return the bicycle to the system. Thus, an empty/full station can create a spill-over of demand to nearby stations. In addition, stations are related by origin-destination pairing. In this paper, we take this effect into consideration for the first time when setting target inventory levels and develop a robust guided local search algorithm for that purpose. We show that neglecting the interactions among stations leads to inferior decision-making.

1. Introduction

Bike Sharing Systems (BSSs) allow customers to rent bicycles at automatic rental stations distributed throughout a city, use them for a short period of time, and then return them to any station. This is an environmentally sustainable mode of transportation and one that can also be integrated with traditional means of public transportation. A significant increase in the number of BSSs and their popularity has recently been seen worldwide (Shaheen et al. (2010)). For a review of the history of BSSs and prospects for their future, see DeMaio (2009) and Shaheen and Guzman (2011).

One of the major aspects affecting the service quality of BSSs is the availability of bicycles and lockers at the different stations, see for example Laporte et al. (2015). Developing an inventory model for a BSS involves unique challenges because of the special features of these systems. A BSS experiences two types of demand: a demand for bicycles, by customers who wish to enter the system (renters), and a demand for lockers, by users who have finished their rides and wish to leave the system (returners). Therefore, basic inventory logic that dictates that a higher inventory level can satisfy more customers is not suitable for addressing this problem. Because each station has a fix capacity, a larger quantity of bicycles at a given station implies a smaller quantity of available lockers. Because of the non-homogeneous asymmetric demand processes that typically characterize BSSs, an inherent imbalance is created, leading to shortages both of bicycles when users are attempting to rent them and of vacant lockers when users are attempting to return them.

To prevent such shortage events, several studies have suggested regulation schemes and policies that influence customer demand to the benefit of the system. For instance, several authors have presented pricing mechanisms that give customers incentives to change their origins and/or destinations, e.g., Chemla et al. (2013b), Pfrommer et al. (2014) and Waserhole and Jost (2016). Kaspi et al. (2014) and Kaspi et al. (2016) proposed a parking reservation policy in which a user reserves a locker at the intended destination station before renting a bicycle, thereby diminishing uncertainty and redirecting that user's demand to an available station. A different kind of policy was presented by Fricker and Gast (2016). Their policy encourages users to choose two destination stations instead of one, and the system then directs them to the station with more vacant lockers.

In practice, the most common approach taken by operators to cope with the difficulties posed by shortages of bicycles or lockers is to reposition bicycles to rebalance the inventory levels at the different stations. This repositioning is typically performed using a fleet of trucks, each of which carries several bicycles. Two types of repositioning can be distinguished: dynamic repositioning and static repositioning. Dynamic repositioning is performed when the system is active to react to the current system state and unexpected events; see Contardo et al. (2012), Kloimüllner et al. (2014) and Pessach (2013). Static repositioning occurs during the night, when traffic is low and the

BSS is idle. The various models and solution methods proposed to address the static repositioning problem include Nair and Miller-Hooks (2011), Benchimol et al. (2011), Chemla et al. (2013a), Raviv et al. (2013), Erdoğan et al. (2014), Angeloudis et al. (2014) and Forma et al. (2015).

Most static repositioning studies assume that a target inventory level or range of inventory levels is known for each station. Only a few studies have addressed the issue of how to determine these target levels for static repositioning. Nair and Miller-Hooks (2011) formulated the problem as a stochastic MIP with the objective of minimizing the cost of the redistribution operation for a required service level. They defined a shortage as a net difference between the total demand over the planning horizon and the total inventory, ignoring the sequence of events occurring in the system. Raviv and Kolka (2013) and Schuijbroek et al. (2017) presented a Markov-chain-based model in which the inventory level is tracked continuously throughout the day. Renters who arrive at an empty station and returners who arrive at a full station are assumed to abandon the system and are considered to be lost sales. Raviv and Kolka (2013) suggested a user dissatisfaction function that measures the performance of a station in terms of the expected penalty due to abandonment by returners and renters as a function of the initial inventory at a single station. Schuijbroek et al. (2017) used dual-bounded service level constraints presented by Nair and Miller-Hooks (2011). Another study that addressed the issue of target inventory levels was conducted by Leurent (2012), who modeled bike sharing stations as a dual Markovian waiting system and assumed that unsatisfied customers would wait at a station rather than abandoning the system. All of these studies considered models based on a single station, meaning that each station's inventory target level was calculated independently of the others and the interactions among stations were neglected. In Vogel et al. (2014), the inventory levels of all stations were set simultaneously, but these authors also ignored the influence of interaction on the system because they treated shortages of bicycles and lockers as lost sales. They determined the stations' inventory levels so as to minimize the total expected operation costs of the system due to relocation while satisfying a minimal level of service.

The interactions among the inventory levels at different stations are an inherent attribute of a BSS. When a customer arrives at an empty station (or when she observes this status online), she can choose between searching for an available bicycle at a neighboring station (referred to as roaming) or abandoning the system to use other means of transportation. Thus, an empty station can create a spill-over of demand to nearby stations. In addition, if the customer decides not to use the system, a future demand for a locker at the destination station is eliminated. Such interactions occur between stations that are not located close to one another. Moreover, when a customer wishes to return her bicycle, she may arrive at a full station and then be *obliged* to find an available space at another station nearby (also referred to as roaming), meaning that a full station will always create locker demand at neighboring stations. In accordance with this concept, Rudloff and Lackner (2013) presented different count demand models for BSSs and demonstrated that full/empty stations have an influence on neighboring stations' demand. George and Xia (2011) addressed the problem of determining the optimal fleet size for a vehicle rental company and derived analytical results for its relationship to vehicle availability at each rental station in the network.

This paper is the first to consider the interactions among stations in BSS, for the purpose of setting target inventory levels. Our contributions are as follows: First, we present a formal definition and mathematical formulation of the BSS inventory problem with station interactions (BSIP-SI). Second, we develop a guided local search algorithm to set the initial inventory level at each station (the target level). This search uses a simulation model in which a user behavior model is implemented. This model includes the roaming between stations (that is, seeking an alternative station) that occurs upon a shortage of bicycles or lockers. Third, we use real data to test our algorithm and compare our results with the common practice of operators and with the results of the model presented in Raviv and Kolka (2013), which ignores these interactions. We show that our algorithm results in a better quality of service for all of the different instances tested. Our results indicate that the interactions among stations' inventory levels cannot be neglected. Specifically, they have an impact on the desired target inventory levels.

The remainder of the paper is organized as follows: in Section 2, we define the problem, the user behavior model, and related assumptions. In Section 3 we present a mathematical formulation of the problem. In Section 4, we characterize the influence of the initial inventory on the system performance and develop our guided local search algorithm accordingly. Section 5 presents the numerical study performed, the properties of the data used, the results and an analysis of the robustness of the search algorithm. Section 6 presents a discussion and summary of the results.

2. Problem Definition

In this section, we provide a formal definition of the bike sharing system inventory problem with station interactions (BSIP-SI). We start with a broad and general definition of the problem. Then, we illustrate some of the more abstract ideas through a more specific formulation that will be used in our numerical experiment in Section 5.

An instance of the problem is defined by the following:

- A set of bike sharing stations - Each station is characterized by its capacity, i.e., the number of lockers/docking poles.
- A general stochastic demand process for desired rides for each origin-destination pair. That is, a ride is a demand for a travel using a bicycle from a certain origin to a certain destination. The origins and destinations are assumed to coincide with the geographic locations of the stations.

The process is defined for a finite planning horizon (typically a working day) and may be non-homogeneous in time and space.

- A journey dissatisfaction function (JDF) with respect to the user. We define a journey to be the itinerary of the user, which brings her from her origin to her destination. A journey may include up to two walking and up to one riding segments. The JDF function maps any combination of a desired ride and a corresponding actual journey to a non-negative value. The ideal journey from station A to B is always the one that proceeds via a direct bicycle ride from A to B, and therefore, the JDF for this scenario is zero by definition. Otherwise, for example, if the user could not find a bicycle at the desired origin and decided to abandon the system or roam to a neighboring station, the JDF returns a larger value that represents the dissatisfaction or dis-utility of the user arising from this occurrence. In our numerical study, we address a special case of the JDF, namely, *excess time*, as will be described later.
- A user behavior model. This model characterizes the choices made by the users, particularly when there are no bicycles at the desired origin station (referred to as a shortage) or when there are no vacant lockers at the desired destination (referred to as a surplus). In general, the user behavior model can be viewed as a decision model that maps a user action to each origin-destination pair and state of the system. The decisions may include waiting for some amount of time at the origin or destination, roaming to a nearby station before renting a bicycle and/or returning it, or abandoning the system and using other modes of transportation. The state of the system at each moment is described by the number of bicycles and (equivalently) the number of available lockers at each station. It is safe to assume that users will strive to minimize their JDF. The general user behavior model is depicted in Figure 1. A detailed example of such a concrete model is given below and depicted in Figure 2.

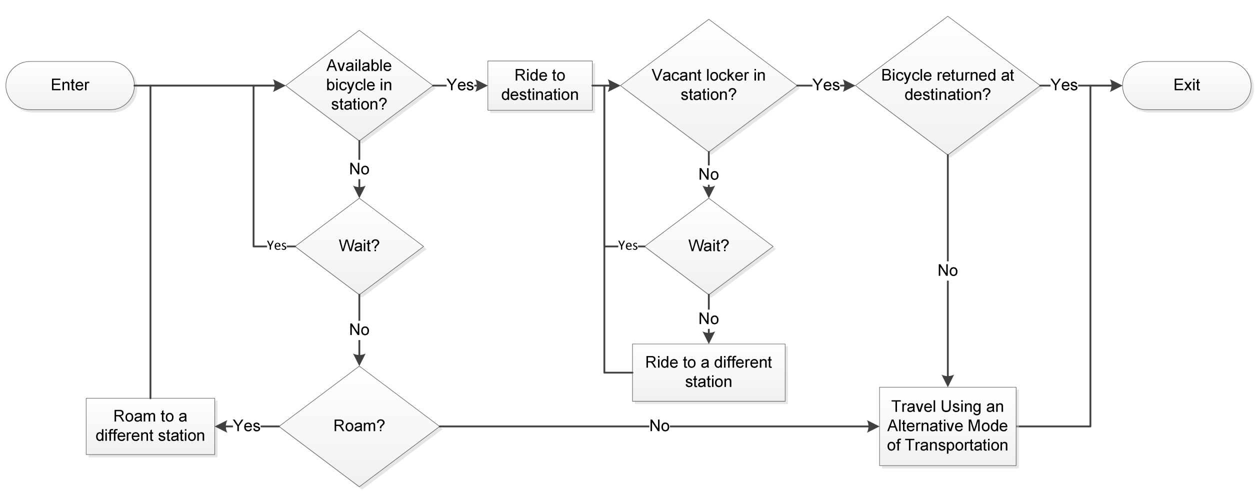


Figure 1 User Behavior Model

Given this input, the BSIP-SI is defined as follows: Set the initial inventory levels of the stations to minimize the total JDF of all journeys over a given planning horizon, typically one day. This problem definition is sufficiently general to capture many assumptions about the preferences and behavior of the users and operators. The use of a given planning horizon is motivated by the fact that in many systems, repositioning work is performed during the night with the intent of preparing the system for the next day. Although (dynamic) repositioning is also performed, during the day, and ideally one would consider the effect of static and dynamic repositioning on each other, dynamic repositioning is out of the scope of this paper. Another underlying assumption of the above problem definition is that the total number of bicycles in the system is not a binding constraint. Indeed, since the cost of a bicycle is relatively low compared with other infrastructural and operational costs of the system, in a well-run BSS, an adequate number of spare bicycles should be available at the operators' disposal at any time.

One example of a JDF, which we consider in the numerical study presented in this paper, is the JDF introduced by Kaspi et al. (2014), i.e., the excess time. The excess time of a journey with respect to a certain ride is defined as the difference between the actual time it takes to complete the journey (the travel time) and the ideal time of the corresponding ride. The actual time of a journey may include waiting and roaming times before and after (or instead of) riding, whereas the ideal time of the corresponding ride refer to a direct bicycle ride between the origin and destination stations. In other words, the excess time reflects any unnecessary time that the user was obliged to spend to reach her desired destination from her origin. This definition of the JDF clearly satisfies the requirement that a value of zero is assigned to ideal itineraries. In addition, it has the virtue of reflecting the extent of the negative implications of each failure in providing the desired service. Operators should take these implications into consideration when setting the inventory levels at stations.

We also adopt the corresponding user behavior model of Kaspi et al. (2014), which is consistent with the excess-time JDF. This user behavior model assumes that each user is independently striving to minimize her own excess time. It also assumes that the users have full information about the state of the system but that they are myopic, that is, at decision points, they do not account for the implications of possible changes in the system state while roaming between neighboring stations in search of available bicycles or vacant docking poles. Moreover, upon renting, they optimistically assume that a vacant docking pole will be available for them at the time of their arrival at the destination.

The following notation is necessary to implement the user behavior model described above:

C_i - Number of lockers at station i, i.e., its capacity

T_{ij} - Travel time by bicycle, i.e., riding time, from station i to station j

W_{ij} - Walking time from station i to station j

$B_i(t)$ - Number of bicycles at station i at time t

Note that $B_i(t)$ is a state variable, unlike the other quantities, which are data parameters.

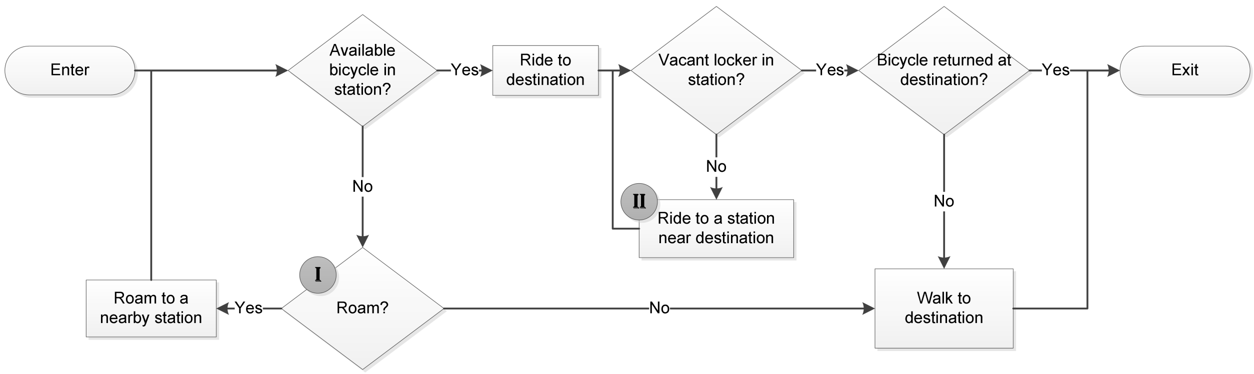


Figure 2 Excess-Time User Behavior Model, adopted from Kaspi et al. (2014)

The model, as depicted in Figure 2, dictates that a user who does not find an available bicycle may choose to roam to a nearby station or walk directly to her destination (I). The user will prefer to rent a bicycle if the total time to reach her destination when that option is chosen is shorter than the walking time to the destination. The total journey time includes the walking time to a non-empty nearby station (at time t) and the riding time from that station to the destination. Here, $k^* = \arg\min_{k:B_k(t)>0}(W_{ik} + T_{kj})$ is the non-empty station to which the user can roam that will result in the shortest total journey time. If $W_{ik^*} + T_{k^*j} < W_{ij}$, then roaming to station k^* is better than walking to the destination and the user will therefore choose to do so; otherwise, she will walk directly to her destination.

Once a bicycle is rented, the user rides to her destination. If, upon arrival at the destination, she finds an available locker, she returns the bicycle there and leaves the system. Otherwise, the user rides to a nearby station with an available locker (at time t), leaves the bicycle there and walks back to the desired destination. The station is chosen in a similar manner: $k^* = \arg\min_{k:B_k(t)<C_k}(T_{jk} + W_{kj})$. If by the time the user arrives at station k^*, say at time t', it appears to be full, a new return station k^{**} is selected such that $k^{**} = \arg\min_{k:B_k(t')<C_k}(T_{k^*k^{**}} + W_{k^{**}j})$. This process is repeated until a vacant locker is found. However, because the availability of vacant lockers is confirmed before the user starts toward the alternative return station, it is most likely that a vacant locker will be found on the first attempt.

The JDF and user behavior model described above abstract out certain considerations of users and operators in BSSs. In particular, other sources of user dissatisfaction due to shortages may exist in addition to excess time. However, these models are sufficiently rich to capture the complex structure of the interactions among stations and thus are useful for setting stations inventory levels. We note that any other JDF that is monotonic and non-decreasing in its occurrences of shortage and surplus events, along with a user behavior model that is consistent with it, can be incorporated into the search algorithm introduced in the next section. For example, one may assume a user behavior model that allows for waiting at the destination station (until a locker becomes available) or using other modes of transportation in addition to walking and riding a bicycle. In such cases, the JDF should reflect the dis-utility associated with these actions. It may include considerations of the uncertainty regarding the total travel time associated with waiting or of the cost of using other modes of transportation.

3. Mathematical formulations of the problem

In this section we provide mathematical formulations of the BSIP-SI when replacing the general stochastic demand process for desired rides with a set of demand realizations that represent it. First we formulate a simplified mixed integer linear programming (MILP) model in which a central planner determines the journey that each user performs in each realization. In the sequel, we modify this model to a bi-level formulation, to account for the decentralized decision making, which better represents practice.

3.1. The centralized model

Input

S Set of stations, indexed by i

R Set of realizations, indexed by r

Q_r Set of users in realization r; each user is represented by a tuple (i, j, t), which means that the user wants to rent a bicycle from station i at time t and ride to station j

J_{qr} Set of possible journeys for user q in realization r, including the possibility of walking from the desired origin to the destination

$\mathcal{T}_{ir}$ Ordered set of all possible epochs in station i of realization r, indexed by t. An epoch is a point in time in which renting or returning a bicycle may occur. In addition, time 0 is defined as the first epoch in each station and realization.

L_{qjr} Travel time of user q in journey j of realiztion r

A_{irt} Set of user-journey pairs (q, j) that end their bicycle ride at station i at epoch $t \in \mathcal{T}_{ir}$

G_{irt} Set of user-journey pairs (q, j) that begin their bicycle ride at station i at epoch $t \in \mathcal{T}_{ir}$

Decision Variables

X_{qjr} Equals 1 if user q takes journey j in realization r

B_{irt} Number of bicycles that arrive to station i in realization r at epoch $t \in \mathcal{T}_{ir}$

H_{irt} Number of bicycles taken from station i in realization r at epoch $t \in \mathcal{T}_{ir}$

I_{irt} Number of bicycles at station i in realization r at epoch $t \in \mathcal{T}_{ir}$

I_i^0 Number of bicycles at station i at epoch 0, in all realizations

$$\min \sum_{r\in R, q\in Q_r, j\in J_{qr}} L_{qjr} X_{qjr} \tag{1}$$

subject to

$$I_{i,r,t-1} + B_{irt} - H_{irt} = I_{irt} \quad \forall i \in S, r \in R, t \in \mathcal{T}_{ir} \tag{2}$$

$$I_{irt} \le C_i \quad \forall i \in S, r \in R, t \in \mathcal{T}_{ir} \tag{3}$$

$$I_i^0 = I_{i,r,0} \quad \forall i \in S, r \in R \tag{4}$$

$$\sum_{j\in J_{qr}} X_{qjr} = 1 \quad \forall r \in R, q \in Q_r \tag{5}$$

$$\sum_{(q,j)\in A_{irt}} X_{qjr} = B_{irt} \quad \forall i \in S, r \in R, t \in \mathcal{T}_{ir} \tag{6}$$

$$\sum_{(q,j)\in G_{irt}} X_{qjr} = H_{irt} \quad \forall i \in S, r \in R, t \in \mathcal{T}_{ir} \tag{7}$$

$$I_i^0 \ge 0, integer \quad \forall i \in S \tag{8}$$

$$I_{irt} \ge 0 \quad \forall i \in S, r \in R, t \in \mathcal{T}_{ir} \tag{9}$$

$$X_{qjr} \in \{0,1\} \quad \forall j \in J_q, r \in R, q \in Q_r \tag{10}$$

The objective function (1) minimizes the total time users spend in the system, by minimizing the sum of the travel times of all users. The total excess time is the difference between the total travel time and the ideal time. Since the ideal time for each journey is a constant, minimizing the total travel time is the same as minimizing the total excess time. Constraints (2) are inventory balance constraints, which keep track of the inventory level at each station, in each realization, at each epoch. Constraints (3) ensure that the inventory at each station is bounded by its capacity, in each realization, at each epoch. Constraints (4) state that the initial inventory level at each station is equal for all realizations. Constraints (5) state that, in each realization, each user is assigned to exactly one journey. Constraints (6) and (7) stipulate that in each realization, for every journey made, there will be bicycles arriving or taken at the relevant stations at the relevant epochs. Constraints (8) define the initial inventory level at each station as nonnegative and integer. Constraints (9) define the number of bicycles at the stations to be nonnegative and (10) define the decisions of which journey each user takes to be binary. Note that since I_i^0 and X_{qjr} are defined to be integer, the values of B_{irt}, H_{irt} and I_{irt} are also integers. The decision variables $I_i^0 \quad \forall i$ represent the solution, where the rest of the decision variables are auxiliary. Note that the initial inventory levels are decided jointly for all the realizations, since this decision is taken before the demand is revealed to the planner. In this formulation, determining all X_{qjr} variables simultaneously, with a single objective function for all users, represents the assumption that a central planner chooses their journeys.

The centralized model provides an assignment of journeys to users that minimizes the total travel time of the users under the given set of realizations. However, since the users make their journey choices independently according to their behavior model, the resulted assignment is not likely to represent the actual choices of the users. Indeed, the central planner may assign a user to a journey which is not convenient for her but reduces the total travel time of all the users in the system. The objective function value of the model is thus a lower bound on the average travel time under any user behavior model.

Unfortunately, it is hard to solve the above MILP model for a realistic system with many stations, large number of possible journeys and a reasonably large set of demand realizations. Therefore, we solved the model with a limited number of possible journeys for each user in each realization, representing the five possible choices that are the most likely to be materialized by the user in that realization, i.e., the ones with the shortest travel times. This set includes in addition the choice of walking between her origin and destination. While the objective value obtained by the model with the limited number of journeys is not a valid lower bound, it can be viewed as an approximation to it. This "approximated" lower bound is used to benchmark our solution method to the decentralized problem. The results are reported in Section 5.5.

Another application of the centralized model could be heuristically using its initial inventory levels as a solution for the BSIP-SI. However, when simulating the system using these levels, it is observed that the resulted total excess time is large . This is demonstrated in Datner (2015).

3.2. The decentralized model

In the decentralized model, the central planner still determines the initial inventory levels I_i^0 $\forall i$, but the decision variables X_{qjr} are determined by the users rather than by the central planner. Each user q aims to choose her journey in a way that would minimize her travel time (and hence her excess time). Thus, this choice depends on the state of the system (the number of bicycles available at each station) and on the user behavior when experiencing shortage or surplus, see the general user behavior model in Figure 1. As for the state of the system - it changes rapidly, due to the stochasticity and complex dynamics of the system (specifically, the interactions between stations), and according to the actual realization. This implies that a complete mathematical formulation, which modifies the centralized formulation to the case of a decentralized system, taking into consideration the user behavior model on top of the above complexities, is unrealistic.

Therefore, we represent the decentralized problem through a bi-level optimization model, in which the lower level decisions are based on the user behavior model. Specifically, define $\mathbb{S}$ to be the state of the system and let $f_q(\mathbb{S})$ be a function that represents the choices made by user q according to the user behavior model, when the state of the system is $\mathbb{S}$. Clearly $\mathbb{S}$ depends on the realization r and the epoch t but we omit these indices for simplicity of the notation. Note that $f_q(\mathbb{S})$ refers to all occurrences in which user q needs to take a decision throughout her journey and the results of these choices determine the entire journey performed, $j \in J_q$. Thus, constraints (5) that state in the centralized problem that each user is assigned to exactly one journey, are replaced by constraints in which the choice of each user q is the above function, $f_q(\mathbb{S})$. That is, constraints (5) are replaced by (for comparison with the centralized problem, we add here the index r):

$$X_{q,f_q(\mathbb{S}),r} = 1 \qquad \forall q \in Q, r \in R \qquad (5'),$$

and the objective function (1) becomes:

$$\min \sum_{r \in R, q \in Q} L_{q,f_q(\mathbb{S}),r} \qquad (1')$$

Note that $L_{q,f_q(\mathbb{S}),r}$ is a function of the state of the system, which depends on the initial state of the system, represented by the decision variables of the central planner, I_i^0 $\forall i$ and the choices of the users dictated by the user behavior model.

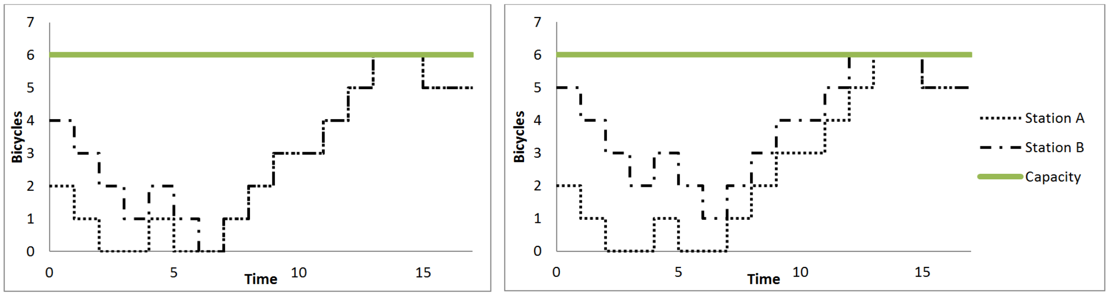
The formulation consisting of (1'), (2)-(4), (5') and (6)-(10) represent the problem faced by the central planner in a decenralized decision making environment. However, due to the decentralized nature of the lower level decisions of the users, this formulation cannot be solved by a general purpose optimization software, and we resort to other methods. Particularly, we use a guided search optimization algorithm, combined with simulation, as described in Section 4.

4. Methodology

Before presenting our algorithm for setting initial inventory levels, we derive a useful property of the inventory dynamics in a single bike sharing station.

Proposition 1 *For a given demand realization at a station, consider the sequence of shortage and surplus events. Let $n \geq 0$ be the number of shortage events that occur before any surplus event. Then, increasing the initial inventory by l bicycles will result in the elimination of at most* $\min(n, l)$ *shortage events.*

Proof Let $B_A(0)$ be the initial inventory at a given station A at the beginning of the planning horizon. Let B be an alternative station facing the same demand realization, with an initial inventory of $B_B(0) = B_A(0) + l$. Consider first the case in which $l \leq n$: After each of the first l shortage



(a) $l \leq n$ (b) $l > n$

Figure 3 Inventory Trajectory at a Single Station

events at station A or surplus events at station B (or any combination of l such events), the difference $B_B(t) - B_A(t)$ is decreased by one. Therefore, after at most l shortage events at station A, $B_A(t) = B_B(t)$, and from this time onward, the inventory levels of the two stations coincide. In other words, there are at most l shortage events that can be eliminated by increasing the initial inventory of a station by l bicycles.

We illustrate the above argument using the example presented in Figure 3a. In this example, $B_A(0) = 2$ and $B_B(0) = 4$, that is, $l = B_B(0) - B_A(0) = 2$. At station A, there are two shortage events (at times 3 and 6) before the first surplus event. With each of these two shortage events, the difference between the two stations is decreased by one, until the station inventories coincide after the second shortage event.

Similarly, consider the case in which $l > n$: After each of the first n shortage events at station A or surplus events at station B (or any combination of n such events), the difference $B_B(t) - B_A(t)$ is decreased by one. Therefore, after at most n shortage events at station A, $B_B(t) - B_A(t) \leq l - n$. Afterward, the inventory levels of the two stations coincide when station A becomes full, which occurs sometime before the first surplus event at station A. In other words, there are at most n shortage events that can be eliminated by increasing the initial inventory of a station by l bicycles; see Figure 3b. □

A similar property also applies for surplus events.

Proposition 2 *For a given demand realization at a station, consider the sequence of surplus and shortage events. Let $n \geq 0$ be the number of surplus events that occur before any shortage event. Then, decreasing the initial inventory by l bicycles will result in the elimination of at most* $\min(n, l)$ *surplus events.*

The proof of Proposition 2 is a mirror image of the proof of Proposition 1 and is thus omitted. An important conclusion that can be drawn from these propositions is that at a station that suffers both surplus and shortage events, only the type of event that occurs first can be directly mitigated by changing the initial inventory level; for example, if the first unmet demand is for a bicycle (i.e., a shortage), then by changing the initial inventory level, we can only prevent shortages and cannot affect any surpluses that occur afterward. We use this observation in the design of our guided local search algorithm by increasing or decreasing the initial inventory levels in accordance with the first type of event observed. We note that Proposition 1 and Proposition 2 are valid only under the assumption of a fixed demand realization at a station.

In reality, any shortage or surplus events at a station affects the demand faced by other stations and may result in a chain reaction throughout the system. The algorithm presented below accounts for this phenomenon by making small changes to the initial inventory levels at each iteration. This in turn, may result in new shortage or surplus event that will be addressed in the next iterations.

We introduce a guided local search algorithm that strives to minimize the total JDF over the planning horizon by setting appropriate initial inventory levels. Our algorithm considers a fixed set of demand realizations, each representing a possible instance of the planning horizon. It searches for the initial inventory levels that minimize the average total JDF over these realizations as an approximation of the expected JDF.

The search is performed iteratively, starting from some initial solution to the problem, i.e., an initial inventory $B_i(0)$ for each station i. In each search iteration, the algorithm estimates the expected total JDF by simulating the system based on a user behavior model, given a set of demand realizations and a vector of initial inventory levels. Information on the occurrences of shortage and surplus events is collected during the simulation. Based on this information, the inventory levels are updated, typically at numerous stations simultaneously. The process is then repeated until some stopping criterion is met. The core of the search algorithm is the mechanism that updates the initial inventory levels of the stations at the end of each iteration.

Guided by Propositions 1 and 2, the information we collect focuses on the first shortage or surplus event at each station. We define the following categories of scenarios:

1. The first shortage event occurs before any surplus event.
2. The first surplus event occurs before any shortage event.
3. No shortage or surplus occurs, but $B_i(t) = 0$ for some t, i.e., the station is empty at some point.
4. No shortage or surplus occurs, but $B_i(t) = C_i$ for some t, i.e., the station is full at some point.

Note that Categories 3 and 4 are not disjoint. In each iteration, we count the number of demand realizations that belong to each category (M_1, M_2, M_3, M_4) for each station. We use these values to determine at which stations a change in the inventory level by one unit may be beneficial, i.e., we apply Propositions 1 and 2 with $l = 1$. An increase in the inventory level could be beneficial at a station where there are more realizations with shortages (Category 1) than surpluses (Category

2). In addition, we must consider the realizations in which there are no shortages or surpluses but the inventory level $B_i(t)$ reaches the station's capacity, that is, the station is full at some point (Category 4). Increasing the inventory level in Category 4 cases will result in a surplus, as in Category 2 cases. Accordingly, we add a bicycle to each station for which $M_1 > M_2 + M_4$. Using the same logic, we remove a bicycle from each station for which $M_2 > M_1 + M_3$. Note that each station can satisfy at most one of the conditions considered above. If a station does not satisfy any of these conditions, this means that it is impossible to reduce the number of shortages or surpluses by changing its inventory level, and therefore, its initial inventory level remains unchanged.

The process is repeated, using the same set of demand realizations, until a solution that was previously considered is encountered. We could stop the search at this point, considering that as a result of its deterministic nature, the algorithm would simply repeat its cycle from this point onward. However, as long as the algorithm's stopping criterion is not met, we instead continue by perturbing the best found inventory levels and continuing from that point. This perturbation also provides some protection against premature convergence to a local minimum. We apply the perturbation by adding a uniform discrete random variable $U[-2,2]$ to the initial inventory level at each station. If this modification results in a solution that exceeds the range $0,...,C_i$ for station i, then the corresponding value is truncated accordingly. Finally, the algorithm stops when a predetermined time budget or number of iterations is reached. A summary of the search stages is illustrated in Figure 4.

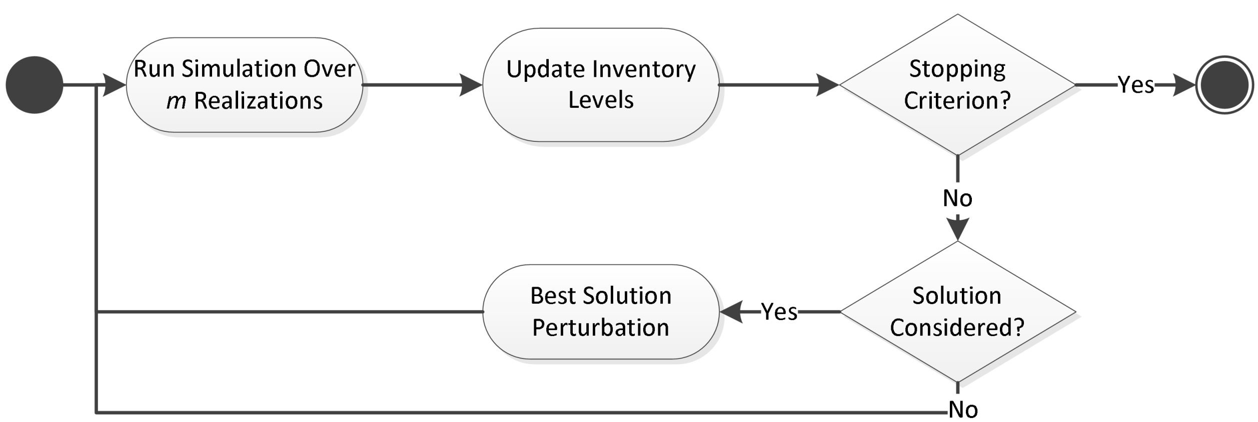


Figure 4 **Search Algorithm**

We refer to the search described thus far as an *occurrence-driven search*. The purpose of this occurrence-driven search is to reduce the number of shortage and surplus occurrences, which is typically consistent with the objective of minimizing any JDF. However, two arbitrary events will not necessarily have the same impact on a JDF. Therefore, it is desirable to devise a search algorithm that prioritizes the elimination of events that will result in a greater effect on the chosen JDF.

Therefore, we introduce a *time-driven search* that is specially tailored for the excess-time JDF. We use an approach similar to that presented above but with an emphasis on the time that users must spend in the system as a result of each avoidable shortage or surplus. Using the same previously described scenario categories, instead of counting the number of realizations, we sum over the avoidable excess time. Let M_1' be the sum of the excess times due to the first shortage event in all realizations of Category 1. This is calculated by, for each such realization, determining the station to which the user roams, $k^* = \arg\min_{k:B_k(t)>0}(W_{ik} + T_{kj})$, and then recording the difference between the resulting journey time with roaming and the ideal time, i.e., $\min(W_{ik^*} + T_{k^*j}, W_{ij}) - T_{ij}$. Similarly, M_2' is the sum of the excess times due to the first surplus event in all realizations in which a surplus event occurs first. This is calculated as $\min_{k:B_k(t)<C_k}(T_{jk} + W_{kj})$.

M_3' is calculated in the same way as M_1' for realizations of Category 3. M_3' is updated at the first time at which the station becomes empty. This represents an evaluation of the excess time that would have been added if the initial inventory level had been reduced by one. Similarly, M_4' is calculated in the same way as M_2' for realizations of Category 4. This represents an evaluation of the excess time that would have been added if the initial inventory level had been increased by one.

Based on the values of $M_1', ..., M_4'$, we update the inventory levels of the stations in the same manner used in the occurrence-driven search: we add a bicycle to each station for which $M_1' > M_2' + M_4'$ and remove a bicycle from each station for which $M_2' > M_1' + M_3'$. The iterations of the search process and the stopping criterion remain the same.

The search algorithm uses a simulation model (described in Section 5.1) that implements the user behavior model using a discrete event simulation architecture. It simulates the system's inventory levels and customers' movement over the planning horizon, given certain initial inventory levels. Different inventory levels can lead to different user decisions, which then lead to different dynamics of the inventory levels, and so on. In this way, the simulation captures the interactions among stations.

5. Numerical Study

In this section, we present a numerical study conducted using the proposed algorithm and its results. Section 5.1 presents the search settings and implementation details. Section 5.2 describes the data used in the study. Section 5.3 reports our results, and Section 5.4 analyzes the robustness of the algorithm.

5.1. Implementation and Experimental Settings

The user behavior model was implemented in a simulation that reproduced two main types of events: renting attempt events (Figure 5) and returning attempt events (Figure 6). In a renting attempt event, a user arrives at a station. If a bicycle is available, a new returning attempt event is created and added to the event queue. Otherwise, based on the logic of the user behavior model, the user either leaves the system or roams to another station. In the latter case, a new renting attempt event is created. In a returning attempt event, a user arrives at a station with a bicycle. If a locker is available, the user leaves the system. Otherwise, the user roams to another station and a new returning attempt event is created.

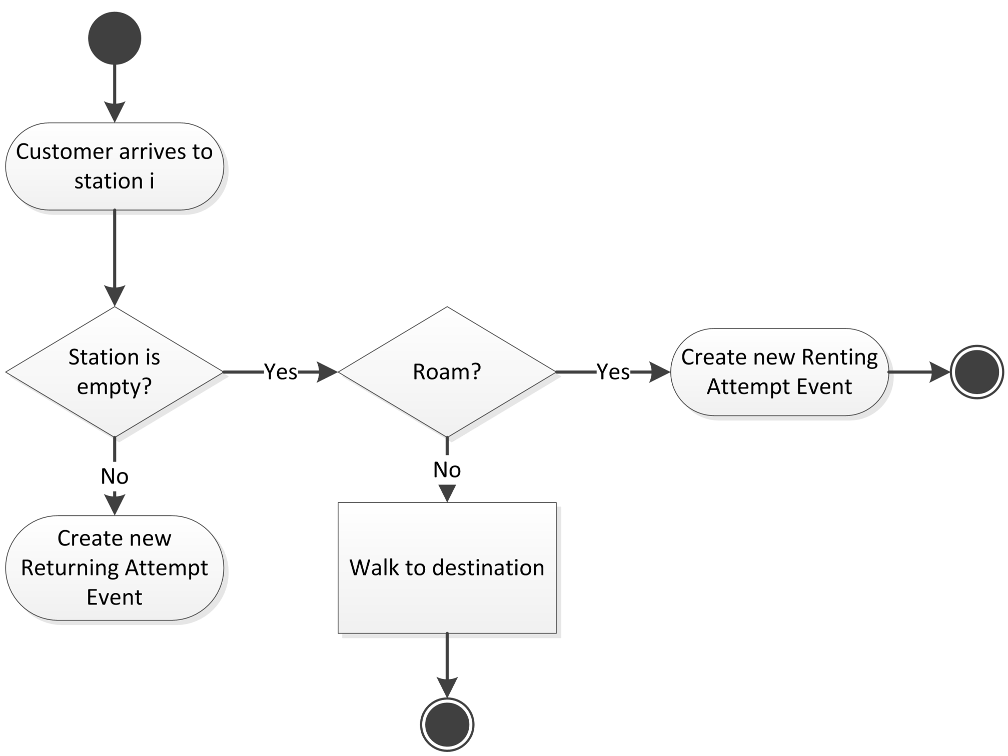


Figure 5 Renting Attempt Event

The two search algorithms and the simulation were coded using MathWorks MATLAB R2011b. The experiments were run on an Intel Xeon X3450 @ 2.67 GHz with 16 GB of RAM. Each of the two search methods was run using three different starting points, i.e., sets of initial inventory levels: (i) Random - a random inventory level at each station; (ii) Half - a starting inventory level at each station equal to half of that station's capacity, a heuristic used both in the literature and in industry; and (iii) R&K - a starting inventory level at each station set using the single-station model suggested by Raviv and Kolka (2013). The stopping criterion was set to 100 iterations.

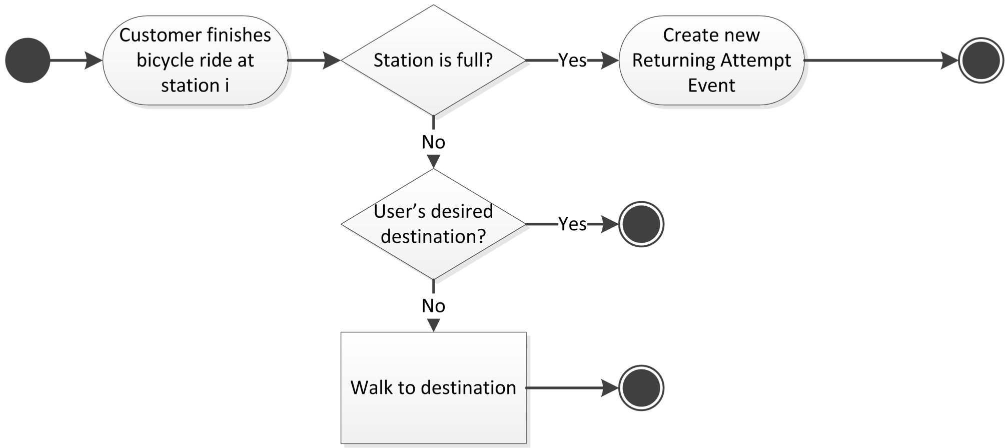


Figure 6 **Returning Attempt Event**

Table 1 **Problem Instances**

	Hubway		Capital		Divvy	
Number of Stations	131		232		300	
Period	May-13	Aug-13	Apr-13	Jun-13	Aug-13	Oct-13
Avg. Rides per Day	3364	4906	7311	8101	4953	5633
Avg. Rides per Planning Horizon	1823	2493	3536	3800	2505	2964

5.2. Input Data

We used data from three BSSs of different sizes, all of which are located in the United States: Hubway in Boston, Capital Bikeshare in Washington, D.C., and Divvy in Chicago. The network topologies of and detailed trip data for these systems are available on their websites. The problem was solved for a planning horizon of 9.5 hours starting at 7:00 am on a working day, assuming without loss of generality that dynamic repositioning would be performed by the end of this planning horizon. For each BSS, we used data from two different months, one working month and one during summer vacation. In this way, we could consider different demand patterns in the same BSS. Several properties of these problem instances are presented in Table 1.

The demand estimation process was executed as described by Kaspi et al. (2014). All rent and return transactions were recorded by the operators. After eliminating holidays and weekend trips, we found that the daily demand patterns did not change significantly throughout each period. By aggregating these transactions over multiple days, we estimated the demand rate of renters for each origin-destination pair during each 30-minute period throughout the day. As may be expected, the demand process was not homogeneous over time. For example, the demand for bicycles at stations located near working areas was low at the beginning of the day and increased significantly toward the end of the working day.

We note that in their current state, the information systems cannot document user abandonments. This is primarily because when a user arrives at an empty station, she will not attempt to rent a bicycle, and therefore, no such attempt is registered by the system. To address this issue, we considered the proportion of time for which a station was empty or full and inflated the demand rates accordingly. However, using the transaction data we obtained, we could not distinguish between users who rented or returned bicycles at their desired origins or destinations and those who were obliged to roam to nearby stations. We note that statistical analysis of this phenomenon will be required to obtain more reliable estimations of demand; however, this is out of the scope of this study.

Based on the estimated origin-destination demand rates, we created a *training set* of 50 realizations, which was used as the input to the search algorithm, and a *test set* of 500 realizations, which was used to evaluate the solutions obtained by the algorithm. In this manner, we simulated the real-life situation in which operators set initial inventory levels based on their forecasts (the training set) and then observe the results on future days (the test set). In addition, the search results showed no effect of over-fitting to the realizations in the training set.

Riding and walking times were estimated using the Google Maps API. For regular trips, it is safe to assume that most users will aim to ride directly from their origins to their destinations. This is not the case for round trips, i.e., trips that begin and end at the same station. The riding time for such trips was set to 30 minutes based on the observed average round-trip travel time.

In summary, our complete data set included riding-time and walking-time matrices, an O-D demand matrix for each 30-minute period of the day, the capacity and location of each station, and demand realizations (i.e., training and test sets) for all six problem instances. These data are available for download from our website at `http://www.eng.tau.ac.il/~talraviv/Publications/`.

5.3. Main Results

In this section, we present our main numerical results. Each problem instance was solved using two search methods (occurrence- and time-driven search) and three starting points. We first note that each of these six solutions consistently outperformed the two alternative solutions with which we compared our results, namely, Half and R&K (note that these solutions should not be confused with the three tested starting points of our search algorithm: Random, Half, and R&K). In Table 2, we report the results for the best solution of the six in each case, referred to as our solution, whereas in Section 5.4, we perform a detailed comparison of all solutions.

The first group of rows presents the total excess time per day (in hours) for the three tested solutions. The second group of rows in Table 2 shows the percentage reduction in excess time achieved by our solution compared with the other two solutions. The third and fourth groups of rows present the number and percentage (with respect to the total demand) of ideal rides, respectively. The average excess time spent in the system by a user who does not have an ideal ride

Table 2 Main Results

		Hubway-May	Hubway-Aug	Capital-Apr	Capital-June	Divvy-Aug	Divvy-Oct
Excess Time (h/day)	Half	85.57	140.43	274.45	318.03	36.08	75.50
	R&K	22.15	54.48	50.92	47.67	14.57	19.12
	Our Search	20.53	49.48	46.08	46.69	14.43	17.35
Excess Time Reduction	Vs. Half	76.01%	64.77%	83.21%	85.32%	60.00%	77.02%
	Vs. R&K	7.35%	9.20%	9.51%	2.08%	0.91%	9.23%
Number of Ideal Rides	Half	1321	1745	2140	2269	2201	2371
	R&K	1623	2119	3104	3207	2385	2754
	Our Search	1644	2133	3120	3377	2384	2767
Ideal Ride Ratio	Half	72.5%	70.0%	60.5%	59.7%	87.9%	80.0%
	R&K	89.0%	85.0%	87.8%	84.4%	95.2%	92.9%
	Our Search	90.2%	85.6%	88.2%	88.9%	95.2%	93.4%
Avg. Excess Time per Non-ideal Ride User (min)	Half	10.2	11.3	11.8	12.5	7.1	7.6
	R&K	6.6	8.7	7.1	4.8	7.3	5.5
	Our Search	6.9	8.2	6.6	6.6	7.2	5.3
Number of Shortage and Surplus Events	Half	630	998	2137	2476	334	748
	R&K	218	437	476	689	125	232
	Our Search	197	413	451	459	127	212
Avg. Number of Shortage and Surplus Events per Non-ideal Ride User	Half	1.3	1.3	1.5	1.6	1.1	1.3
	R&K	1.1	1.2	1.1	1.2	1.0	1.1
	Our Search	1.1	1.1	1.1	1.1	1.1	1.1
Avg. Running Time (h)		1.02	1.34	2.06	2.08	1.49	1.69

is presented in the fifth group of rows. The sixth group of rows shows the total number of shortage and surplus occurrences. Note that this number is slightly different from the number of non-ideal rides because the same user may experience one or more shortage and/or surplus events. In the seventh group of rows, the average number of shortage and/or surplus occurrences per non-ideal ride user is presented. The last row reports the average running time of the search algorithm per problem instance.

We observe that our solution consistently outperforms the other solutions in terms of total excess time. The excess time reduction is statistically significant compared with both the Half and the R&K solution methods. The mean excess time, when calculated based on the 500 realizations in the test set, is significantly smaller for our solution. In a paired t-test the p-value is at most 0.0006. Interestingly, for the same BSS in different months, the percentage reductions in excess time achieved by our solution can be very different, as in the cases of Capital and Divvy. Recall that the R&K solution considers only a single station at a given time, neglecting the interactions among stations. Therefore, in instances in which such interactions are rare because of the balanced nature of the demand process, it is more difficult to affect the total excess time merely by adjusting the initial inventory levels.

Although we advocate the use of the excess-time JDF, we recognize that many other authors and operators use other performance measures, particularly the number of shortage and surplus occurrences. We observe in Table 2 that minimizing the total excess time results in reducing the number of such shortage or surplus events in five of the six cases. In one case (Divvy-Aug), the number of these events in our solution is slightly larger compared with that in the R&K solution. Similarly, the number and ratio of ideal rides are typically larger in our solution. Another interesting observation is related to the average excess time spent by users who do not have an ideal ride. In most cases, in addition to reducing the number of users who experience non-ideal rides, the average excess time they spend is also reduced. Furthermore, the average number of shortage and surplus occurrences experienced by a non-ideal ride user is no larger than in the other solutions. In short, our solution results in a higher number of satisfied users, and most unsatisfied users are less discomforted in terms of both the number of shortage and surplus events and their consequent excess time.

Next, let us consider the results of the occurrence- and time-driven search algorithms. In Table 3, we compare the solutions that represent the best results (among the three starting points) achieved by each of these algorithms in terms of excess time. The values presented in the table represent the difference between the two solutions, where positive values in the table correspond to higher measures for the time-driven search. The first column shows the names of the problem instances. In the second column, we present the percentage reduction in excess time achieved by the time-driven search minus the corresponding value for the occurrence-driven search. The third column shows the average difference in the number of shortage and surplus events per user between these two solutions. Similarly, in the last column, we present the difference in the ideal ride ratio. Recall that unlike the two previous measures, the ideal ride ratio is a measure that should be maximized; thus, negative values here reflect better results in the time-driven search.

As expected, the time-driven search algorithm is better suited to minimizing the total excess time. Interestingly, the two algorithms yield very similar results in terms of the number of shortage and surplus occurrences and the ideal ride ratio, although the time-driven algorithm demonstrates a slight advantage. We conclude that the excess time may be a good surrogate objective function for various service quality measures. To further investigate the properties of the search method, in the following subsection, we focus only on the time-driven algorithm.

5.4. Robustness of the Algorithm

In this section, we consider the effects of different starting points and different training sets on the performance of the time-driven algorithm. In Table 4, we show how the search is affected by the different starting points (i.e., Random, Half and R&K). For each problem instance and starting

Table 3 Time-driven Results Minus Occurrence-driven Results

	Excess Time Reduction	No. Shortage and Surplus per User	Ideal Ride Ratio
Hubway-May	0.81%	-0.0006	-0.0011
Hubway-Aug	1.09%	0.0040	-0.0004
Capital-Apr	0.44%	-0.0002	-0.0003
Capital-June	0.33%	0.0025	-0.0008
Divvy-Aug	-0.05%	0.0096	-0.0004
Divvy-Oct	1.72%	0.0086	-0.0007

Table 4 Robustness to the Starting Point

	Random	Half	R&K
Hubway-May	7.00% The quality of the solutions	**7.35%**	6.91%
Hubway-Aug	**9.20%**	8.85%	8.80%
Capital-Apr	9.27%	9.21%	**9.51%**
Capital-June	1.67%	1.97%	**2.08%**
Divvy-Aug	0.48%	**0.91%**	0.80%
Divvy-Oct	8.96%	9.11%	**9.23%**

point, the table presents the excess time improvement compared with the R&K solution (as in the fifth row of Table 2).

The most important observation to be drawn from Table 4 is that our search algorithm is not highly sensitive to its starting point, which is advantageous. Recall that the three starting points that we used were a randomly generated vector, a vector representing half of the capacity at each station and the solution obtained using the R&K method. Each of these starting points could itself represent a solution to the problem; among them, Random is typically the worst and R&K is the best in terms of excess time. Interestingly, the table shows that a better starting point does not necessarily lead to a better solution. In fact, the R&K starting point led to the best final result in only half of the problem instances. Clearly, if sufficient computational resources are available, some improvement may be gained by running the algorithm with multiple starting points, including various random vectors.

Next, let us examine the sensitivity of the algorithm to the specific training set of 50 realizations that was used as the input to the search algorithm. We created three more such sets based on the same demand processes and ran the search again using the R&K starting point. The solutions were evaluated using the same test set of 500 realizations as was the solution of the original search. The results are displayed in Table 5. The first column provides the names of the problem instances. The second column gives the excess time improvement (over R&K) of the original training set's solution for the R&K starting point. The remainder of the columns show the improvement rates achieved using the three other training sets with the same starting point. It is evident that the

Table 5 Robustness to the Training Set

	Original	1	2	3
Hubway-May	6.91%	6.81%	6.91%	6.81%
Hubway-Aug	8.80%	8.70%	8.65%	8.75%
Capital-Apr	9.51%	7.58%	9.01%	9.30%
Capital-June	2.08%	1.59%	2.42%	2.10%
Divvy-Aug	0.80%	0.80%	1.12%	1.20%
Divvy-Oct	9.23%	9.66%	10.26%	10.68%

search algorithm is fairly robust. In nine of the eighteen runs, the improvement achieved using the newly generated training sets was equal to or larger than the original. Therefore, there is no reason to suspect that the search was over-fitted to the original training set.

5.5. Approximated lower bound

In this section we present the results of a numerical experiment that we conducted with the centralized MILP model presented in Section 3. Our goal was to calculate an approximated lower bound (A-LB) obtained by this model and compare it with the actual excess time simulated with initial inventory levels based on our time driven search. Both the MILP model and the search procedure were applied with the same 50 demand realizations (the training set). The actual excess time was estimated by a different set of 500 demand realizations drawn from the same stochastic process.

We used the same six datasets as in Section 5. The results of this experiment are summarized in Table 6. Each row refers to one of the instances. In the first column the instance name is presented. In the next four columns the excess time obtained from the centralized model, the time driven search procedure, the optimal solution obtained from Raviv and Kolka (2013) (R&K) and the naive approach of setting the inventory level to “half” are presented, respectively. In the sixth column the relative optimality gap between the solution of the time driven search and the approximated lower bound is presented. The gap is calculated as the ratio between the absolute optimality gap and the upper bound. In the two right most columns we present the relative share of the gap closed by the time driven search heuristic as compared to the R&K and the naive solutions.

It is apparent from Table 6 that either the A-LB provided by the centralized model is rather weak or the solution obtained by our time driven search procedure is far from the optimum. We believe that the former is true. Indeed, when looking closely into the solution of the centralized planner, we observe that many of the decisions made by the model are far from being optimal from the individual user perspective. These decisions are done by the centralized model only for the “greater good” of all users. Moreover, this model exploits full information about future demand, which is not available in practice. On the positive side, we observe that the time driven search closes most

Table 6 Analyses of the A-LB obtained by the centralized model

	Centralized model (A-LB)	Time driven Search	R&K	Half	Optimality Gap	Gap closed from R&K	Gap closed from Half
Hubway-May	9.72	20.53	22.15	85.57	52.7%	13.0%	85.7%
Hubway-Aug	19.19	49.48	54.48	140.43	61.2%	14.2%	75.0%
Capital-Apr	25.02	46.08	50.92	274.45	45.7%	18.7%	91.6%
Capital-June	24.73	46.69	47.67	318.03	47.0%	4.3%	92.5%
Divvy-Aug	5.88	14.43	14.57	36.08	59.2%	1.6%	71.7%
Divvy-Oct	6.75	17.35	19.12	75.50	61.1%	14.3%	84.6%
Average					54.5%	11.0%	83.5%

of the optimality gap between the naive solution and the approximated lower bound, and a non-negligible part of gap between the R&K solution and the approximated lower bound. This further strengthen our claim that advanced optimization techniques has the potential of improving the service level provided by a BSS without using additional physical resources, such as repositioning vehicles.

6. Conclusions

In this paper, we introduced the problem of setting the initial inventory levels in a BSS with station interactions and developed a simulation-based guided local search algorithm that optimizes the quality of service. Our algorithm is novel in the sense that it extracts information from the dynamics observed in the simulation. We proved that only the first shortage or surplus event at each station in each demand realization can be eliminated by changing the initial inventory level at that station by one unit. We used this property to guide our search procedure. The algorithm is capable of capturing and considering complex interactions in the system that originate from the behavior of the users. Such complexities could not be addressed without the use of simulation. The effectiveness of our algorithm was demonstrated using actual demand data from three real BSSs.

In our model, it is assumed that the goal of the operator when setting the initial inventory levels is to minimize the JDF, which is equivalent to maximizing the quality of service. A legitimate criticism of this modeling assumption is that the operator may have other objectives, such as minimizing his operational cost and, in particular, the cost of repositioning bicycles between stations. Moreover, it is not always possible to satisfy the inventory levels prescribed by our model. This can be the case, for example, because of the capacity and time constraints of the repositioning operation. Therefore, it is important to also explore values of solutions in the neighborhood of the solution obtained by our algorithm. Such an investigation is out of the scope of the current study and will be an important topic for future research.

Our numerical study shows that the interactions among stations should not be neglected when planning the inventory levels of BSS stations, as done by previous authors, e.g., Raviv and Kolka

(2013). In most of the cases, our guided search algorithm saves about 7-9% of the excess time, compared with the R&K solution. We believe, that such an improvement in the service level, achieved without additional physical resources, is worth the effort of looking into the interactions between the stations. However, if the system is not very loaded or the demand is relatively balanced, the interactions are infrequent and thus it is less crucial to take them into account.

In any transportation system, and particularly in a BSS, each user is selfishly attempting to minimize her own dissatisfaction by selecting the best possible itinerary. If a central planner could assign an itinerary to each user, the total JDF could be reduced much further, although certain users might be worse off. A model that is based on a central planner that assigns users to journeys is formulated as a mixed integer linear program. This model determines the optimal initial inventory level with respect to a large set of demand realizations. However, when we used the inventory levels prescribed by this model in combination with the simulation and user behavior model described in this paper, we found that the resulting excess time and number of shortage and surplus events were not competitive with our results or even with the R&K solution. This finding can be attributed to the gap between the itineraries that would be selected by a central planner and those selected by the users themselves. The results of this numerical experiment is reported in Datner (2015).

The discussion above is relevant to various decisions regarding the design and operation of BSSs, e.g., repositioning operations and the locations and capacities of stations. Future research should consider the behavior of users and interactions among stations when devising models for these problems. For example, when operators are considering the trade-off between setting up many small stations or fewer stations with greater capacity, the corresponding problem cannot be correctly solved without considering that users can roam between stations.

References

Angeloudis, Panagiotis, Jun Hu, Michael GH Bell. 2014. A strategic repositioning algorithm for bicycle-sharing schemes. *Transportmetrica A: Transport Science* **10**(8) 759–774.

Benchimol, Mike, Pascal Benchimol, Benoît Chappert, Arnaud De La Taille, Fabien Laroche, Frédéric Meunier, Ludovic Robinet. 2011. Balancing the stations of a self service bike hire system. *RAIRO-Operations Research* **45**(01) 37–61.

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

Chemla, Daniel, Frédéric Meunier, Thomas Pradeau, Roberto Wolfler Calvo, Houssame Yahiaoui. 2013b. Self-service bike sharing systems: simulation, repositioning, pricing. Technical report.

Contardo, Claudio, Catherine Morency, Louis-Martin Rousseau. 2012. Balancing a dynamic public bike-sharing system. CIRRELT Technical report.

Datner, Sharon. 2015. Setting inventory levels in a bike sharing network. Tech. rep., MS.c. Thesis, Tel-Aviv University.

DeMaio, Paul. 2009. Bike-sharing: History, impacts, models of provision, and future. *Journal of Public Transportation* **12**(4) 3.

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

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

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

George, David K., Cathy H. Xia. 2011. Fleet-sizing and service availability for a vehicle rental system via closed queueing networks. *European Journal of Operational Research* **211**(1) 198 – 207.

Kaspi, Mor, Tal Raviv, Michal Tzur. 2014. Parking reservation policies in one-way vehicle sharing systems. *Transportation Research Part B: Methodological* **62** 35–50.

Kaspi, Mor, Tal Raviv, Michal Tzur, Hila Galili. 2016. Regulating vehicle sharing systems through parking reservation policies: Analysis and performance bounds. *European Journal of Operational Research* **251**(3) 969–987.

Kloimüllner, Christian, Petrina Papazek, Bin Hu, Günther R Raidl. 2014. Balancing bicycle sharing systems: An approach for the dynamic case. *Evolutionary Computation in Combinatorial Optimisation*. Springer, 73–84.

Laporte, Gilbert, Frédéric Meunier, Roberto Wolfler Calvo. 2015. Shared mobility systems. *4OR* **13**(4) 341–360.

Leurent, Fabien. 2012. Modelling a vehicle-sharing station as a dual waiting system: stochastic framework and stationary analysis. Working paper.

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

Pessach, Dana. 2013. Dynamic repositioning in a bike sharing system. Tech. rep., MS.c. Thesis (in Hebrew), Tel-Aviv University.

Pfrommer, Julius, Joseph Warrington, Georg Schildbach, Manfred Morari. 2014. Dynamic vehicle redistribution and online price incentives in shared mobility systems. *Intelligent Transportation Systems, IEEE Transactions on* **15**(4) 1567–1578.

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

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

Rudloff, Christian, Bettina Lackner. 2013. Modeling demand for bicycle sharing systems–neighboring stations as a source for demand and a reason for structural breaks. *Transportation Research Board Annual Meeting*.

Schuijbroek, Jasper, Robert Hampshire, Willem-Jan van Hoeve. 2017. Inventory rebalancing and vehicle routing in bike sharing systems. *European Journal of Operational Research* **257**(3) 992–1004.

Shaheen, Susan, Stacey Guzman. 2011. Worldwide bikesharing. *Access Magazine* **1**(39).

Shaheen, Susan, Stacey Guzman, Hua Zhang. 2010. Bikesharing in europe, the americas, and asia: past, present, and future. *Transportation Research Record: Journal of the Transportation Research Board* (2143) 159–167.

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

Waserhole, Ariel, Vincent Jost. 2016. Pricing in vehicle sharing systems: Optimization in queuing networks with product forms. *EURO Journal on Transportation and Logistics* **5**(3) 293–320.