MIT Open Access Articles

Improving Customer Satisfaction in Bike Sharing Systems through Dynamic Repositioning

Improving Customer Satisfaction
in Bike Sharing Systems through Dynamic Repositioning

Supriyo Ghosh[1*] , **Jing Yu Koh**[2*] and **Patrick Jaillet**[3]

[1]IBM Research Center, Singapore 018983

[2]Singapore University of Technology and Design, Singapore

[3]Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology

supriyog@ibm.com, jingyu_koh@mymail.sutd.edu.sg, jaillet@mit.edu

Abstract

In bike sharing systems (BSSs), the uncoordinated movements of customers using bikes lead to empty or congested stations, which causes a significant loss in customer demand. In order to reduce the lost demand, a wide variety of existing research has employed a fixed set of historical demand patterns to design efficient bike repositioning solutions. However, the progress remains slow in understanding the underlying uncertainties in demand and designing proactive robust bike repositioning solutions. To bridge this gap, we propose a dynamic bike repositioning approach based on a probabilistic satisficing method which uses the uncertain demand parameters that are learnt from historical data. We develop a novel and computationally efficient mixed integer linear program for maximizing the probability of satisfying the uncertain demand so as to improve the overall customer satisfaction and efficiency of the system. Extensive experimental results from a simulation model built on a real-world bike sharing data set demonstrate that our approach is not only robust to uncertainties in customer demand, but also outperforms the existing state-of-the-art repositioning approaches in terms of reducing the expected lost demand.

1 Introduction

The extensive usage of private vehicles significantly contributes to major growing concerns such as global warming, air pollution, usage of non-renewable resources, traffic congestion and emission of greenhouse gases [Wright and Fulton, 2005]. To mitigate these concerns, Bike Sharing Systems (BSSs) have become increasingly ubiquitous in recent years as they can provide an attractive and green alternative to private transportation. Thanks to the boom of the sharing economy, more than 1,700 BSSs are already deployed in the major cities of the world and around 390 programs are in the planning stage [Meddin and DeMaio, 2018]. A few popular

examples of BSSs are *Capital Bikeshare* in Washington DC, *Hubway* in Boston, *CitiBike* in New York, *Bixi* in Montreal, *Wuhan* and *Hangzhou Public Bicycle* in China, etc.

In a typical dock-based BSS, a set of stations is strategically placed throughout the city, each of which has a fixed docking capacity. Due to involuntary and uncoordinated one-way trips of customers using the bikes, the stations often get imbalanced over time which causes either starvation (supply of bikes is lower than the demand) or congestion (supply of bikes is higher than the demand) of bikes at stations, leading to a significant amount of unsatisfied customers. The starvation of bikes at stations causes monetary loss to the BSS operators and in the worst case, the unsatisfied customers might leave the system forever. In order to reduce the number of unsatisfied customers (referred to as lost demand), many BSS operators employ carrier vehicles (e.g., medium size trucks) to rebalance the entire system at the end of the day, which is known as static repositioning [Chemla *et al.*, 2013]. In addition, several BSS operators also reposition bikes during the day by either myopically matching the producer and consumer stations [Contardo *et al.*, 2012; Schuijbroek *et al.*, 2017; Lowalekar *et al.*, 2017] or by considering a fixed set of historical demand points [Shu *et al.*, 2013; Ghosh *et al.*, 2017; Mellou and Jaillet, 2019], which is referred to as dynamic repositioning. However, the real-world spatio-temporal customer demand for many BSSs (specially for densely populated cities) is highly uncertain and varies significantly over time and therefore, the methods with fixed demand may fail significantly in meeting the future demand. [Ghosh *et al.*, 2016] propose a two-player iterative game approach between the repositioning planner and an adversary to develop a robust repositioning approach by considering the variance in the historical demand. However, as the proposed adversarial game is designed to tackle the worst-case scenarios, it may fail to reduce the expected lost demand adequately and therefore, customer satisfaction level might degrade.

In order to tackle the aforementioned concerns and to improve customer satisfaction, we explore new advances in data uncertainty and propose a probabilistic model for dynamic repositioning of bikes by taking into account the underlying uncertainty in spatio-temporal customer demand. We employ a tractable probabilistic satisficing model, referred to as the *T-model* [Jaillet *et al.*, 2016]. The principal idea behind the *T-model* is to identify an efficient solution which maximizes the

probability of satisfying a decision criterion under data uncertainty. Specifically, a satisficing decision criterion is used to evaluate how well a given solution would remain feasible under uncertainty. We employ this reasoning behind the *T-model* for dynamic repositioning of bikes, where the objective is to maximize the probability of meeting the uncertain customer demand. In a nutshell, we first identify the spatio-temporal demand over multiple days from the historical demand data and develop a novel mixed integer linear program (MILP) model that maximizes the log-likelihood of meeting all the customer demand while ensuring the physical routing constraints of the carrier vehicles.

We develop an online approach for dynamic repositioning of bikes which uses the current inventory information (i.e., distribution of bikes) of stations and carrier vehicles and the future demand information for the next time step. Once the repositioning solution is identified, we execute the solution on a real-world simulator to identify the station inventory level for the next time step. This iterative process continues until we reach the final time step. Lastly, we perform extensive experimental studies on a real-world bike sharing data set from Boston city, and show that our approach significantly reduces the expected lost demand over a wide range of state-of-the-art benchmark approaches. In addition, our approach is shown to be robust to uncertainty in future demand.

2 Related Work

Due to the proliferation of BSSs, several bike repositioning models have been proposed recently which can be broadly categorized into three threads of research. The first thread of research is based on static bike repositioning problem (SBRP) where the movements of bikes by customers during the decision period is assumed to be negligible. [Chemla *et al.*, 2013] solve the SBRP with a single vehicle by employing a branch and cut algorithm. [Raviv *et al.*, 2013] propose an MILP that produces the bike repositioning activities and routing solutions for a set of vehicles to maximize satisfaction of the customers. [Schuijbroek *et al.*, 2017] model the bike flow network with Markov assumptions and use closed-form expressions to predict bounds on the expected demand at each station. Rebalancing is performed to meet the maximum and minimum service bounds by dividing the stations into clusters and assigning one vehicle to each cluster. However, as the system evolution and station inventory levels change with uncertain customer demand, these approaches are not suitable for solving our problem during the day.

The second thread of research focuses on dynamic repositioning that takes into consideration the flows of bikes by customers during the repositioning period. Several recent research papers [Pfrommer *et al.*, 2014; Singla *et al.*, 2015; Ghosh and Varakantham, 2017] propose myopic incentivization mechanisms for encouraging users to support in rebalancing the system. [Shu *et al.*, 2013] propose an offline optimization model by considering the future expected demand for a long period to deal with future demand surges. [Ghosh *et al.*, 2015; 2017] jointly consider the repositioning of bikes and the routing problem of vehicles with predicted long-term future demand and present decomposition and ab-

straction mechanisms to speed up the solution process. [Mellou and Jaillet, 2019] propose methodologies for accurate demand estimation and provide efficient optimization models for dynamic repositioning with expected future demand which is solved within a real-time framework using station clustering based decomposition method. In contrast to these papers which consider future expected demand and avoid the underlying uncertainties in demand scenarios, our approach takes into account the uncertainties associated with historical demand scenarios while generating a dynamic repositioning solution so as to improve the overall customer satisfaction.

The last thread of research focuses on data-driven robust repositioning and optimization. To tackle the uncertainties in spatio-temporal customer demand, data-driven solution approaches have been proposed for many applications including emergency medical response [Ghosh and Varakantham, 2016; 2018; Konda *et al.*, 2018], taxi fleet optimization [Lowalekar *et al.*, 2018] and healthcare operation [He *et al.*, 2015]. However, the progress remains slow in designing data-driven robust bike repositioning solutions. [Ghosh *et al.*, 2016] propose a robust and online bike repositioning approach using a scenario generation based two-player iterative game. While the proposed game-theoretic approach outperforms existing dynamic repositioning approaches, it is primarily designed to tackle the worst-case scenarios and therefore, it might fail to reduce the expected lost demand and to improve the customer satisfaction level adequately.

3 Model and Background

In this section, we provide a generic model for the **D**ynamic **r**outing and **R**ep**O**sitioning of **B**ikes under demand **U**ncertainty using **Sa****T**isficing approach (**DrROBUST**). In addition, we briefly introduce the satisficing approach and the probabilistic notion of meeting customer demand which form the cornerstone of our solution methodology.

3.1 Model: DrROBUST

We extend the generic model of dynamic routing and repositioning problem (DRRP) introduced by [Ghosh *et al.*, 2015; 2017] for defining our problem. DrROBUST is compactly represented using the following tuple:

$$< \mathcal{S}, \mathcal{V}, \boldsymbol{C}^{\#}, \boldsymbol{C}^{*}, d^{\#,0}, \{\sigma_v^0\}, \boldsymbol{P}, \boldsymbol{F} >$$

We have a set of base stations $\mathcal{S}$ (strategically placed throughout the city) and a set of carrier vehicles $\mathcal{V}$. $C_s^{\#}$ denotes the finite docking capacity of station $s \in \mathcal{S}$ and C_v^* denotes the capacity (i.e., the number of available bike slots) of vehicle $v \in \mathcal{V}$. $d^{\#,0}$ represents the initial distribution of bikes at the stations. σ^0 symbolizes the initial distribution of vehicles at stations. Specifically, $\sigma_v^0(s)$ is fixed to 1 if vehicle v is initially present at station s and is set to 0 otherwise. $\boldsymbol{P}$ is a two-dimensional matrix that provides the relative distance between two stations. $\boldsymbol{F}$ denotes a set of K demand scenarios which are used for training purposes, where $F_{s,s'}^k$ represents the customer demand for a given time step for scenario k that originates from station s and plans to reach station s' in the next time step.

In the similar direction of [Ghosh *et al.*, 2016], we make the following assumptions for the ease of representation and

evaluation, which can be relaxed with minor modifications to our approach: (a) Bike trips by the customers are always completed within one time step. That is to say, a customer who reserves a bike at time step t always returns the bike at the destination station at the beginning of time step $t + 1$. However, a vehicle can visit multiple stations within one time step; (b) Customers are impatient in nature. Therefore, if the origin station of a customer is empty, he leaves the system instead of visiting the nearest station. While returning a bike, if the destination station is full, then he always returns the bike to the nearest available station; and (c) The events at each time step follow a particular sequence. The customers first return their bikes at the destination station, then the repositioning of bikes by the vehicles are carried out and lastly, the arrival customers pick up their bikes.

3.2 Satisficing Approach

Satisficing is a decision-making approach under data uncertainty which was introduced by [Simon, 1959]. The satisficing approach aims to achieve solutions that satisfy the problem's constraints as well as possible in case of uncertain environments. [Charnes and Cooper, 1963] first use the satisficing approach within an optimization framework. To overcome the inherent scalability issues of satisficing based optimization methods, [Jaillet et al., 2016] recently proposed a class of tractable probabilistic satisficing models. The proposed tractable model can be formally defined as follows:

$$
\begin{aligned}
\max \; & \rho(\boldsymbol{\alpha}) \\
\text{s.t. } & \boldsymbol{A}(\boldsymbol{z})\boldsymbol{x} \geq \boldsymbol{b}(\boldsymbol{z}) \; \forall \boldsymbol{z} \in \mathcal{U}(\boldsymbol{\alpha}) \qquad (1) \\
& \boldsymbol{x} \in \mathcal{X} \\
& \boldsymbol{\alpha} \in \mathcal{S}
\end{aligned}
$$

where $\boldsymbol{x}$ is a decision variable of dimension N and $\boldsymbol{z}$ is a K dimensional uncertain variable that influences the entries of function $\boldsymbol{A} : \mathbb{R}^K \to \mathbb{R}^{M \times N}$ and $\boldsymbol{b} : \mathbb{R}^K \to \mathbb{R}^M$. $\mathcal{U}(\boldsymbol{\alpha})$ denotes the family of uncertainty sets parameterized by the variable $\boldsymbol{\alpha}$ with support set $\mathcal{S}$. Therefore, $\boldsymbol{z}$ checks the feasibility constraints over the uncertainty set $\mathcal{U}(\boldsymbol{\alpha})$. $\rho(\boldsymbol{\alpha})$ is a generic function that determines the level of satisfaction expected from the optimized solution. For instance, $\rho(\boldsymbol{\alpha}) = \boldsymbol{\alpha}$ indicates that the model should identify the most robust solution $\boldsymbol{x}$ that would remain feasible in the problem's constraints when $\boldsymbol{z}$ arises from maximally sized uncertainty set. Intuitively, the optimization model (1) maximizes a function defined over the uncertain parameters indexed over the support set while ensuring that the chosen uncertain parameters satisfy a set of feasibility constraints.

Probability of Meeting Demand
We now provide the motivation behind adopting the satisficing model in the context of bike sharing system. We consider a set of historical demand data sets to learn the underlying uncertainty in demand. Specifically, we consider customer demand for K days and for each day we extract the demand information for each time step and for each pair of source and destination station. Motivated by [Jaillet et al., 2016], we assume that the uncertain demand at station s at a given time step (i.e., z_s) is independent but not necessarily identically

distributed variable with support W_s. The support set for station s at a given time step is defined as a set of integer values, $W_s = \{\zeta_s^l, \dots, \zeta_s^{L(s)}\}$. For instance, if we consider 4 days of historical demand for training purposes and the demand at station s at time step t for those 4 days was 2, 1, 0 and 1, then the support set for station s at that time step is defined as $W_s = \{0, 1, 2\}$. Let, $\lambda_s^l = P(z_s \leq \zeta_s^l)$ denote the realization probability that the demand at station s in a given time step is bounded by ζ_s^l. Specifically, λ_s^l is an approximation of the cumulative distribution function (CDF) of z_s. Intuitively speaking, if we are able to meet ζ_s^l demand at station s, then the probability of meeting all the demand in the station is λ_s^l according to the historical demand data. In case of the previous example, the historical data indicates a customer demand of 0 and 2 bikes on one day and a demand of 1 bike on two days. Hence, $\lambda_s^0 = P(\widetilde{z}_s \leq 0) = \frac{1}{4}$; $\lambda_s^1 = P(\widetilde{z}_s \leq 1) = \frac{3}{4}$; and $\lambda_s^2 = P(\widetilde{z}_s \leq 2) = 1$. Given the support set and realization probability, our goal is to identify a bike repositioning solution that maximizes the sum of log likelihood of meeting the demand over all the stations for a given decision period (i.e., $\sum_s \log(P(z_s \in W_s))$).

4 Solution Approach for DrROBUST

We propose a rolling horizon framework for solving the DrROBUST, where the following two components are executed sequentially at each time step:

- Given the distribution of bikes at stations, generate repositioning solution for the next time step;

- Simulate the repositioning solution along with customer movements to generate the bike distribution at each station for the next time step.

4.1 Generate Repositioning Solution

In this section, we describe the method for computing a bike repositioning solution that maximizes the customer satisfaction level where the demand is uncertain. To generate the bike repositioning solution, we provide a novel mixed integer linear program (MILP). At a given decision epoch t, the distribution of bikes at stations (i.e., $d^{\#,t}$) is given as an input to the MILP model. In addition, we precompute the values of $\boldsymbol{\lambda}$ and $\boldsymbol{\zeta}$ from the historical demand data for the planning period and augment them into the inputs of the optimization model. It should be noted that a vehicle can visit multiple stations within one decision epoch. Let us assume that a vehicle can visit a maximum of R stations within one decision epoch. To represent the sequence of moves for a vehicle, we use another time index (also referred to as episode) $r \in \{0, \dots, R\}$. We now introduce a set of decision and intermediate variables to formally represent the optimization model. Let $y_{s,v}^{+,r}$ and $y_{s,v}^{-,r}$ denote the number of bikes picked up and dropped off by vehicle v from station s at episode r, respectively. $z_{s,v}^r$ denotes the routing decision variable which is set to 1 if vehicle v is stationed at station s at episode r and 0 otherwise. Let $d_v^{*,r}$ denote the number of bikes present in vehicle v at episode r and $D_{s,s'}^{v,r}$ represent the distance travelled by vehicle v during episode r if it moves from station s to s'.

As indicated earlier, our goal is to maximize the log likelihood of satisfying the uncertain customer demand which can

take any value from the support set $\boldsymbol{W}^1$. Therefore, we introduce the binary variables $\boldsymbol{\alpha}$ to capture the realization of customer demand from the uncertainty set. Specifically, α_s^l is set to 1 if $\zeta_s^l \in W_s$ is selected as the demand bound to be met for station s and otherwise it is set to 0. Intuitively, since the support set W_s is ordered non-decreasingly and the realized demand can only take one value from W_s for the station s, our goal is to set the value α_s^l to 1 for the highest possible ζ_s^l given that the physical routing and repositioning constraints are ensured. Therefore, for a given station s, the log likelihood of meeting the realized demand can be represented as:

$$\log(P(\widetilde{z}_s \in W_s)) = \sum_{l \in [L(s)]} \alpha_s^l \log(\lambda_s^l)$$

The optimization model for generating the repositioning solution is represented compactly in Table (1). Our objective (delineated in Expression (2)) is to maximize the log likelihood of meeting the realized demand over all the stations. The constraints associated with this repositioning task generation problem are described as follows:

The supply of bikes is sufficient for realized demand. Constraints (3) ensure that the supply of bikes is higher than the selected demand bound (i.e., $\sum_{l \in L(s)} \alpha_s^l \zeta_s^l$) so that we can always satisfy the realized demand which is lower than the chosen demand bound. The supply of demand is computed as the sum of the bikes present in the station (i.e., $d_s^{\#,t}$) and the net amount of dropped off bikes by all the carrier vehicles during the planning period (i.e., $\sum_{r,v}(y_{s,v}^{-,r} - y_{s,v}^{+,r})$). Constraints (4) enforce that exactly one demand bound is chosen from the support set for each station by allowing only one of the decision variable of $\boldsymbol{\alpha}_s$ to be set to 1. While generating a repositioning solution, it is possible to encounter an unexpected situation where any of the previously observed demand cannot be satisfied. For example, at a given decision period, consider a station which has experienced at least one demand in all the historical demand data. But, while generating the repositioning solution, we observe that the station is empty and due to physical routing limitation of the vehicles, it is impossible to reposition bikes to the station during the planning period. To avoid such infeasibility, we introduce a slack variable ρ_s as the dummy supply of bikes at station s in constraints (3). Constraint (5) enforces that the cumulative value of slack variables over all the station is bounded by a given input value ρ. As the aforementioned unexpected situation arises rarely, we manually tune the input parameter ρ. We begin with $\rho = 0$ and increase the value of ρ gradually by 1 until we identify a feasible solution, which ensures that we minimize the number of unsatisfied customer demand in case of an unexpected situation. As the value of ρ is always fairly low and the MILP solvers (e.g. CPLEX) promptly returns an infeasibility error, a feasible value of ρ can be found quickly.

The flows of bikes in and out of vehicles are preserved. Constraints (6) enforce the flow preservation of bikes on vehicles. The initial load of bikes in vehicles (i.e., $\boldsymbol{d^{*,0}}$) is given as input to the optimization model. So, constraints (6) ensure

$$\max \sum_{s \in S} \sum_{l \in [L(s)]} \alpha_s^l \log(\lambda_s^l) \tag{2}$$

$$\text{s.t.} \quad \sum_{l \in [L(s)]} \zeta_s^l \alpha_s^l \leq d_s^{\#,t} + \sum_{r,v}(y_{s,v}^{-,r} - y_{s,v}^{+,r}) + \rho_s, \ \forall s \tag{3}$$

$$\sum_{l \in [L(s)]} \alpha_s^l = 1, \qquad \forall s \tag{4}$$

$$\sum_{s \in S} \rho_s \leq \rho \tag{5}$$

$$d_v^{*,r} = d_v^{*,r-1} + \sum_s (y_{s,v}^{+,r} - y_{s,v}^{-,r}), \qquad \forall v,r \tag{6}$$

$$y_{s,v}^{+,r} + y_{s,v}^{-,r} \leq C_v^* \cdot z_{s,v}^r, \qquad \forall s,v,r \tag{7}$$

$$\sum_{r,v} y_{s,v}^{+,r} \leq d_s^{\#,t}, \qquad \forall s \tag{8}$$

$$\sum_{r,v} y_{s,v}^{-,r} \leq C_s^{\#} - d_s^{\#,t}, \qquad \forall s \tag{9}$$

$$\sum_{s \in S} z_{s,v}^r = 1, \qquad \forall v,r \tag{10}$$

$$D_{s,s'}^{v,r} \geq P_{s,s'}(z_{s',v}^r + z_{s,v}^{r-1} - 1), \quad \forall s,s',v,r \tag{11}$$

$$u \sum_{r,s,s'} D_{s,s'}^{v,r} + m \sum_{r,s}(y_{s,v}^{+,r} + y_{s,v}^{-,r}) \leq Q, \ \forall v \tag{12}$$

$$0 \leq d_v^{*,r}, y_{s,v}^{+,r}, y_{s,v}^{-,r} \leq C_v^*, D_{s,s'}^{v,r} \geq 0 \tag{13}$$

$$\alpha_s^l \in \{0,1\}, z_{s,v}^r \in \{0,1\} \tag{14}$$

Table 1: SOLVEDRROBUST($\boldsymbol{\zeta}, \boldsymbol{\lambda}, t, \boldsymbol{d^{\#}}, \boldsymbol{d^{*,0}}$)

that the number of bikes present at vehicle v at the episode r (i.e., $d_v^{*,r}$) is equivalent to the sum of the number of bikes present in the vehicle at the previous episode (i.e., $d_v^{*,r-1}$) and the net number of picked up bikes from stations during that episode (i.e., $\sum_s(y_{s,v}^{+,r} - y_{s,v}^{-,r})$).

A vehicle can only pickup/drop-off bikes from a station if it is currently present there. Constraints (7) enforce that the number of bikes picked up or dropped off by vehicle v from station s at episode r is bounded by the vehicle capacity if v is present at station s (i.e., $z_{s,v}^r = 1$) and 0 otherwise.

Total number of bikes picked up or dropped off from a station is less than the available bikes or open docks. As multiple vehicles can pick up bikes from the same station within the planning period, constraints (8) enforce that the total number of picked up bikes from station s by all the vehicles over all the episodes is bounded by the number of bikes present in the station (i.e., $d_s^{\#,t}$). Similarly, constraints (9) assure that the total number of dropped-off bikes by all the vehicles at station s is bounded by the number of empty docks at the station (i.e., $C_s^{\#} - d_s^{\#,t}$).

Total time spent by a vehicle in traveling and repositioning of bikes is less than the duration of planning period. Constraints (10) ensure that at a given episode r, vehicle v should be placed in one of the stations by allowing only one

of the routing variables to be set to 1 (i.e. $\sum_s z_{s,v}^r = 1$). Constraints (11) compute a lower bound on the distance travelled by vehicle v if it travels from station s to s' during episode r (i.e., $D_{s,s'}^{v,r}$). $D_{s,s'}^{v,r}$ is set to $P_{s,s'}$ if the vehicle travels from station s to s' during episode r (i.e., $z_{s,v}^{r-1} = z_{s',v}^r = 1$) and otherwise it is set to 0. Let u denote the unit for converting distance to time, m denote the time required to pickup/drop-off one bike and Q denote the duration of planning period. Constraints (12) ensure that the total time spent by vehicle v in routing (i.e., $u \sum_{r,s,s'} D_{s,s'}^{v,r}$) and pickup/drop-off events (i.e., $m \sum_{r,s}(y_{s,v}^{+,r} + y_{s,v}^{-,r})$) is bounded by the duration of planning period, Q. So, these constraints enforce an upper bound on the routing distance travelled by a vehicle in each time-step. Therefore, as shown in section 5, the cumulative routing distance for our approach is always highly competitive to other benchmark approaches.

Vehicle capacity is not exceeded. Constraints (13) assure that the number of picked up or dropped off bikes in aggregate is bounded by the vehicle's capacity.

We further exploit the structure of the MILP of Table (1) with the following observation which helps to improve the computational efficiency of the proposed MILP.

Observation 1 *The integrality of $\boldsymbol{\alpha}$ variables in constraints* (14) *can be relaxed to* $0 \leq \alpha_s^l \leq 1$, *without compromising the feasibility or optimality of the optimization problem delineated in Table* (1).

Proof: The objective function (2) computes a point-wise maximum of linear functions of $\boldsymbol{\alpha}$ and therefore, convex in $\boldsymbol{\alpha}$ [Boyd and Vandenberghe, 2004]. As a convex function over a compact domain achieves the optimal solution at an extreme point of the feasible region, we can relax the binary variables $\boldsymbol{\alpha}$ to continuous ones. Let l^* denote the index of the largest $\boldsymbol{\lambda}_s$ value that satisfies all the feasibility constraints for station s. Then, any convex combination of $\boldsymbol{\lambda}_s$ with continuous values of $\boldsymbol{\alpha}_s$ (i.e., $\sum_{l \in [L(s)]} \alpha_s^l \log(\lambda_s^l)$) is less than or equal to $\log(\lambda_s^{l^*})$. Hence, in the optimal solution, $\alpha_s^{l^*}$ should be set to 1 even if $\boldsymbol{\alpha}$'s are defined as continuous variables. ∎

4.2 Simulation Model

We now describe a data-driven simulation model (adapted from [Ghosh *et al.*, 2016]) which is used to execute the bike repositioning strategies and for evaluating their performance on testing demand scenarios. Let $f_{s,s'}^t$ represent the number of customers who intend to travel from station s to s' at time step t. Let $d_s^{\#,t}$ denote the number of bikes stationed at s after the repositioning tasks are completed for time step t. As the number of available bikes changes due to repositioning, the actual flows of bikes between stations are computed as follows: (a) If the arrival demand at a station is less than the supply of bikes, then all the customers are served; (b) If the arrival demand at a station is higher than the supply of bikes, then the actual flow of bikes (denoted as $x_{s,s'}^t$) is determined based on the relative ratio $\frac{f_{s,s'}^t}{\sum_{s'} f_{s,s'}^t}$ as shown in equation (15).

$$x_{s,s'}^t = \left\{ \begin{array}{ll} f_{s,s'}^t & \textbf{if } \sum_{s'} f_{s,s'}^t \leq d_s^{\#,t} \\ \frac{f_{s,s'}^t}{\sum_{\tilde{s}} f_{s,\tilde{s}}^t} \cdot d_s^{\#,t} & \textbf{otherwise} \end{array} \right\} \quad (15)$$

Once we compute the actual flows of bikes between the stations, the distribution of bikes at station s for the next time step $t+1$ is determined as the sum of un-hired bikes at time step t, the net amount of incoming bikes and the net amount of dropped off bikes at station s by the vehicles.

$$d_s^{\#,t+1} = d_s^{\#,t} + \Big[\sum_{\tilde{s}} x_{\tilde{s},s}^t - \sum_{s'} x_{s,s'}^t\Big] + \Big[Y_s^{-,t+1} - Y_s^{+,t+1}\Big] \quad (16)$$

It should be noted that equation (16) ignores the station capacity constraints. To tackle such boundary conditions, we transfer excess bikes (i.e., $d_s^{\#,t+1} - C_s^{\#}$) to the nearest available station. In the experimental results, we report these excess numbers as the lost demand at the destination station. This bike distribution (i.e., $d^{\#,t+1}$) is then used to compute the repositioning strategy for the next time step. We continue this iterative process until the last time step is reached.

5 Experimental Results

We evaluate the performance[2] of our approach with respect to the key performance metric of loss in demand on a real-world data set from *Hubway*[3]. The *Hubway* data set contains the following information: (1) Customer trip records, from which we compute the demand scenarios; (2) The number of stations, their capacity and initial distribution of bikes at each of the stations; (3) Geographical locations of stations, from which we calculate the relative distance between two stations; and (4) The number of vehicles and their capacity. The *Hubway* system consists of 95 base stations and 3 vehicles. We consider 6 hours of planning horizon in the morning peak (6AM-12PM) which is divided into 12 decision epochs, each having a duration of 30 minutes. We generate 60 demand scenarios for the weekdays from three months of historical trip data. As the historical trip data ignores the unobserved lost demand, we employ a micro-simulation model from [Ghosh *et al.*, 2016] with one minute of time discretization to determine the period when a station was empty and inject artificial demand based on the observed demand at that station in previous time step. We observe that the mean of total pickup/return demand for a single day over 6 hours of planning period is 1485. From 60 demand scenarios, 20 scenarios are used for training purposes and other 40 scenarios are used for testing. We compare the utility of our approaches with the following four benchmark approaches:

Static repositioning: The vehicles are used for repositioning at the end of the day to achieve a predefined inventory level (e.g., 50% of the capacity) for the stations. For this baseline approach, we simulate the customer flow without any repositioning of bikes during the planning period.

Offline repositioning: For the offline approach [Shu *et al.*, 2013], the repositioning strategy is generated once at the beginning for the entire planning horizon by considering the expected future demand (computed from the training scenarios) to better account for future demand surges.

---

[2]All the linear optimization models are solved using IBM ILOG CPLEX Optimization Studio V12.7 on a 2.1 GHz Intel Xeon E5 machine with 16 cores and 32GB RAM.

[3]Data is taken from *Hubway* bike sharing company of Boston [http://hubwaydatachallenge.org/trip-history-data]

	Lost demand at the time of bike pickup			Lost demand at the time of bike return			Routing Distance
	MEAN	MAX	STDEV	MEAN	MAX	STDEV	(KM)
Static	283.6	459	79.7	65.8	103	19.3	0
Online	272.4	436	78.3	50.1	87	21.22	250.1
Offline	205	356	70.4	54.8	116	18.03	282.2
Robust	201.4	337	64.5	<u>49.2</u>	81	<u>14.05</u>	217.7
DrROBUST	<u>160.9</u>	<u>317</u>	62.4	52	<u>79</u>	15.61	243.2

(a) Comparison results with 3 episodes in each decision epoch ($R = 3$).

	Lost demand at the time of bike pickup			Lost demand at the time of bike return			Routing Distance
	MEAN	MAX	STDEV	MEAN	MAX	STDEV	(KM)
Static	283.6	460	79.7	65.8	103	19.4	0
Online	263.7	416	71.7	54.1	85	20	282.6
Offline	217.8	367	68.9	81	159	21.7	486.5
Robust	178.7	304	58.3	<u>53.3</u>	105	16.8	259.9
DrROBUST	<u>131.9</u>	<u>261</u>	55.2	54.4	<u>81</u>	<u>15.8</u>	336.5

(b) Comparison results with 4 episodes in each decision epoch ($R = 4$).

Table 2: Lost demand statistics on the *Hubway* data set.

Online repositioning: For the online heuristic [Schuijbroek *et al.*, 2017], our objective is to bound the inventory level within 10% of the mean demand (learnt from training data), while ensuring the physical routing limitations of vehicles.

Robust repositioning: We employ a scenario generation based two-payer game approach from [Ghosh *et al.*, 2016] to generate the robust repositioning strategy. In each iteration of the game, a worst demand scenario is generated to counter the repositioning strategy of the current iteration and then the repositioning planner produces a robust strategy that minimizes the maximum lost demand over all the previously generated worst-case scenarios.

Performance comparison. We summarize the key performance statistics of DrROBUST against all the benchmark approaches in Table 2. We provide statistics for two types of loss in demand: (a) lost demand occurred at the time of bike pickup due to starvation of bikes at stations; and (b) lost demand occurred at the time of bike drop-off due to congestion of bikes at stations. Table 2(a) summarizes the lost demand statistics at the pickup and drop-off time for different approaches with 3 episodes (i.e., a vehicle is allowed to visit maximum 3 stations in each of the decision epochs). Due to the uncertainties in customer demand, the *Robust* approach outperforms all the other benchmarks (i.e., *Static*, *Offline* and *Online* approaches). Our proposed approach performs better than the *Robust* approach both in terms of reducing the expected and worst-case lost demand. On average, our approach reduces the overall lost demand (i.e., sum of lost demands at the pickup and drop-off time) by 15% and the worst-case lost demand is also reduced by around 5% over the *Robust* approach. Similar performance statistics are shown in Table 2(b) where we allow a vehicle to visit 4 stations in each decision epoch. We observe a consistent pattern in results that our approach reduces the expected lost demand by at least 19% over all the benchmark approaches. Moreover, our approach reduces the worst-case lost demand by at least 9% and therefore, it is robust to the uncertainty in demand.

Routing cost comparison. The next set of results provide the routing cost comparison. Table 2 reports the average cumulative distance (which is directly proportional to the routing cost) travelled by 3 vehicles over the entire planning hori-
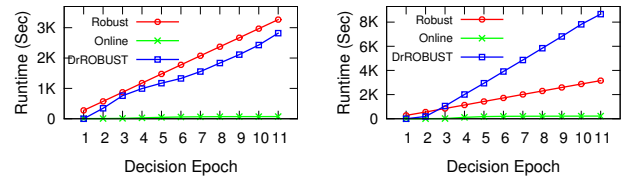

Figure 1: (Cumulative) Runtime comparison on *Hubway* data set for: (a) $R = 3$; and (b) $R = 4$.

zon. As expected, vehicles travel the maximum distance for the *Offline* approach to minimize future demand surges. Although both *DrROBUST* and *Robust* employ similar physical routing constraints for vehicles, the average routing cost for *DrROBUST* is higher (albeit by a small amount) than the *Robust* approach. However, as *DrROBUST* significantly reduces the lost demand over the *Robust* approach, it can contribute to higher profits for the BSS operators.

Runtime performance. In the last set of results, we show the runtime performance of our approach on the real-world demand scenarios. For fairness in comparison, we only provide runtimes for *Online*, *Robust* and *DrROBUST* approaches as all of them generate decisions in each time step in a round-robin fashion. Figure 1(a) depicts the runtime performance with 3 episodes where in the X-axis we vary the number of decision epochs and the Y-axis denotes the cumulative runtime. While the *Online* approach performs the best, our approach was always faster than the *Robust* approach. Figure 1(b) delineates the runtime with 4 episodes. Although our approach takes the longest time to generate a better quality solution in comparison to other benchmarks, the runtime is always bounded by 15 minutes in each decision epoch.

6 Concluding Remarks

In this paper, we propose a probabilistic satisficing based optimization approach to solve the dynamic repositioning problem in bike sharing systems. We develop a computationally efficient optimization model for maximizing the probability of satisfying the customer demand by exploiting the underlying uncertainties learnt from the historical demand. The empirical results on a real-world bike sharing data set demonstrate that our approach is robust to demand uncertainty and outperforms existing best known repositioning approaches in reducing the expected lost demand and therefore, improves the overall customer satisfaction. In future, this work can be extended to an offline planning and online execution setting where the repositioning strategy is generated in each time step by considering the demand uncertainties for multiple future time steps to better account for future demand surges.

References

[Boyd and Vandenberghe, 2004] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[Charnes and Cooper, 1963] Abraham Charnes and William W Cooper. Deterministic equivalents for optimizing and satisficing under chance constraints. *Operations research*, 11(1):18–39, 1963.

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

[Contardo *et al.*, 2012] Claudio Contardo, Catherine Morency, and Louis-Martin Rousseau. Balancing a dynamic public bike-sharing system. Technical report, CIRRELT, 2012.

[Ghosh and Varakantham, 2016] Supriyo Ghosh and Pradeep Varakantham. Strategic planning for setting up base stations in emergency medical systems. In *ICAPS*, pages 385–393. AAAI Press, 2016.

[Ghosh and Varakantham, 2017] Supriyo Ghosh and Pradeep Varakantham. Incentivizing the use of bike trailers for dynamic repositioning in bike sharing systems. In *ICAPS*. AAAI Press, 2017.

[Ghosh and Varakantham, 2018] Supriyo Ghosh and Pradeep Varakantham. Dispatch guided allocation optimization for effective emergency response. In *AAAI*, pages 775–783, 2018.

[Ghosh *et al.*, 2015] Supriyo Ghosh, Pradeep Varakantham, Yossiri Adulyasak, and Patrick Jaillet. Dynamic redeployment to counter congestion or starvation in vehicle sharing systems. In *ICAPS*. AAAI Press, 2015.

[Ghosh *et al.*, 2016] Supriyo Ghosh, Michael Trick, and Pradeep Varakantham. Robust repositioning to counter unpredictable demand in bike sharing systems. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2016.

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

[He *et al.*, 2015] Shuangchi He, Melvyn Sim, and Meilin Zhang. Data-driven patient scheduling in emergency departments: A hybrid robust-stochastic approach. *Available at Optimization-Online http://www. optimization-online. org/DB_HTML/2015/11/5213. html*, 2015.

[Jaillet *et al.*, 2016] Patrick Jaillet, Sanjay Dominik Jena, Tsan Sheng Ng, and Melvyn Sim. Satisficing awakens: Models to mitigate uncertainty. *Available at Optimization-Online http://www.optimization-online.org/DB_HTML/2016/01/5310.html*, 2016.

[Konda *et al.*, 2018] Muralidhar Konda, Supriyo Ghosh, and Pradeep Varakantham. Reserved optimisation: Handling incident priorities in emergency response systems. In *International Conference on Automated Planning and Scheduling (ICAPS)*, pages 330–338, 2018.

[Lowalekar *et al.*, 2017] Meghna Lowalekar, Pradeep Varakantham, Supriyo Ghosh, Sanjay Dominic JENA, and Patrick Jaillet. Online repositioning in bike sharing systems. In *ICAPS*. AAAI Press, 2017.

[Lowalekar *et al.*, 2018] Meghna Lowalekar, Pradeep Varakantham, and Patrick Jaillet. Online spatio-temporal matching in stochastic and dynamic domains. *Artificial Intelligence*, 261:71–112, 2018.

[Meddin and DeMaio, 2018] Russell Meddin and Paul DeMaio. The bike sharing world map. http://www.bikesharingworld.com, 2018. Accessed: 2018-07-09.

[Mellou and Jaillet, 2019] Konstantina Mellou and Patrick Jaillet. Dynamic resource redistribution and demand estimation: An application to bike sharing systems. *Available at SSRN: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3336416*, Feb 2019.

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

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

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

[Shu *et al.*, 2013] Jia Shu, Mabel C Chou, Qizhang Liu, Chung-Piaw Teo, and I-Lin Wang. Models for effective deployment and redistribution of bicycles within public bicycle-sharing systems. *Operations Research*, 61(6):1346–1359, 2013.

[Simon, 1959] Herbert A Simon. Theories of decision-making in economics and behavioral science. *The American economic review*, 49(3):253–283, 1959.

[Singla *et al.*, 2015] Adish Singla, Marco Santoni, Gábor Bartók, Pratik Mukerji, Moritz Meenen, and Andreas Krause. Incentivizing users for balancing bike sharing systems. In *AAAI*, pages 723–729, 2015.

[Wright and Fulton, 2005] Lloyd Wright and Lewis Fulton. Climate change mitigation and transport in developing nations. *Transport Reviews*, 25(6):691–717, 2005.