Incentivizing the Use of Bike Trailers for Dynamic Repositioning in Bike Sharing Systems

Supriyo Ghosh
School of Information Systems
Singapore Management University
supriyog.2013@phdis.smu.edu.sg

Pradeep Varakantham
School of Information Systems
Singapore Management University
pradeepv@smu.edu.sg

Abstract

Bike Sharing System (BSS) is a green mode of transportation that is employed extensively for short distance travels in major cities of the world. Unfortunately, the users behaviour driven by their personal needs can often result in empty or full base stations, thereby resulting in loss of customer demand. To counter this loss in customer demand, BSS operators typically utilize a fleet of carrier vehicles for repositioning the bikes between stations. However, this fuel burning mode of repositioning incurs a significant amount of routing, labor cost and further increases carbon emissions. Therefore, we propose a potentially self-sustaining and environment friendly system of dynamic repositioning, that moves idle bikes during the day with the help of bike trailers. A bike trailer is an add-on to a bike that can help with carrying 3-5 bikes at once. Specifically, we make the following key contributions: (i) We provide an optimization formulation that generates "repositioning" tasks so as to minimize the expected lost demand over past demand scenarios; (ii) Within the budget constraints of the operator, we then design a mechanism to crowdsource the tasks among potential users who intend to execute repositioning tasks; (iii) Finally, we provide extensive results on a wide range of demand scenarios from a real-world data set to demonstrate that our approach is highly competitive to the existing fuel burning mode of repositioning while being green.

Introduction

Due to its potential to mitigate the carbon emissions and traffic congestion, Bike Sharing Systems (BSSs) have been widely adopted in major cities across the world. According to Meddin and DeMaio (2016), 1139 systems with a fleet of over 1,445,000 bicycles are already installed in major cities and additionally, 357 systems are either in planning stages or under construction. Popular examples of BSS are *Capital Bikeshare* in Washington DC, *Hubway* in Boston, *Bixi* in Montreal, *Vélib'* in Paris, *Wuhan and Hangzhou* Public Bicycle in Hangzhou etc. In a regular BSS, base stations are scattered throughout a city and each station is stocked with a pre-determined number of bikes at the beginning of the day. According to personal needs, users with membership card can pickup and drop-off bikes at any base station, each of which has a finite number of docks. At the end of the work day, carrier vehicles (e.g., trucks) are used to rebalance the entire system so as to return to some pre-determined configuration at the beginning of the day.

BSSs often experience a significant loss in customer demand during the day due to the uncoordinated movements of customers. Moreover, BSS companies (e.g., *Vélib'* in Paris) are often penalized by local governments for loss in customer demand (Schuijbroek, Hampshire, and Van Hoeve 2017), as it can result in usage of fuel burning modes of private transport. To address this problem, a wide variety of research papers and current systems employ the idea of repositioning idle bikes with the help of vehicles during the day, by taking into account the movements of bikes by customers. While such a method of repositioning can help reduce imbalance, there are multiple drawbacks: (a) Vehicles incur substantial routing and labor costs; (b) More importantly, the fuel burning model of repositioning is at odds with the environment friendly nature of BSSs; and (c) Finally, due to a limited number of these vehicles, they are typically not sufficient to account for all the lost demand.

As an alternative, some BSS operators (e.g., *CitiBike* in NYC) have recently introduced the notion of bike trailers (O'Mahony and Shmoys 2015). A bike trailer is an add-on to a bike that can carry a small number of bikes (e.g., each bike trailer can hold 3-5 bikes) and is useful to relocate bikes to nearby stations. Trailers are an environment friendly mode of repositioning the bikes. Existing research by O'Mahony and Shmoys (2015) has focussed on computing the repositioning tasks for trailers with the assumption that dedicated staff can execute the repositioning tasks. However, given the limited budget availability, it is not economically viable to employ dedicated staff for each of the trailers.

This paper introduces a potentially self-sustaining repositioning system that addresses this Dynamic Repositioning and Routing Problem with Trailers (DRRPT). We employ an unique combination of optimization and mechanism design that crowdsources the repositioning tasks to the potential users while working within the budget constraints of the operator. Specifically, we provide a rolling horizon framework, where at each time step we have two components executed one after another:

1. We first employ mixed integer linear optimization to generate potential repositioning tasks along with their valuations at the next time step.

2. We employ an incentive compatible mechanism to crowdsource (using payment/trip based incentives) the repositioning tasks to the users who are interested in executing those tasks within the budget constraints of the operator.

There has been existing research (Singla et al. 2015; Pfrommer et al. 2014) that has focussed on providing incentives to users for assisting with repositioning. However, this line of work has primarily focussed on individual bikes and has taken a myopic (individual station) view on whether a bike is required at a station. In this work, we provide an end to end system that takes the global view (all stations) of the repositioning requirements and incentives their execution within the budget constraints.

We evaluate our system using a simulation model which is built on the realized demand scenarios from a real-world data set. At each time step the two components of the rolling horizon framework are executed on this simulator to identify the distribution of bikes for the next time step. This iterative process continues until we reach the last time step. Experimental results on multiple synthetic and a real-world data set of *Hubway* (Boston) BSS demonstrate that our approach is highly competitive in terms of reducing the expected lost demand, over the fuel burning model of repositioning.

Related Work

Given the practical importance of BSSs, repositioning problem has been studied extensively in the literature. We categorize existing research into three threads: (a) Static and (b) Dynamic repositioning using carrier vehicles; (c) Incentivizing customers and utilizing trailers for repositioning.

Static repositioning is the problem of finding routes for a fleet of vehicles to reposition bikes at the end of the day when the movements of bikes by customers are negligible, to achieve a pre-determined inventory level at the base stations. Chemla, Meunier, and Wolfler Calvo (2013) employ branch and cut algorithm to solve the static repositioning problem with more than a hundred stations. Benchimol et al. (2011) propose an approximate solution for the static rebalancing and routing problem with a single vehicle using insights from the solution of C-delivery TSP (Chalasani and Motwani 1999). Raviv, Tzur, and Forma (2013) and Raviv and Kolka (2013) provide scalable approximate solutions for multiple vehicles using mathematical optimization models where they design an objective function that penalizes unavailability of bikes or empty docks. Di Gaspero, Rendl, and Urli (2013; 2015) employ constraint programming (CP) to efficiently solve the static repositioning problem using large neighbourhood search. As the uncertainty and changes in demand alter the station inventory level, these static approaches are not suitable to solve our problem during the day.

Dynamic repositioning is referred to as the case when the movements of customers during the day are considered in the planning period. Nair and Miller-Hooks (2011) and Nair et al. (2013) provide a dynamic repositioning approach by employing dual-bounded joint-chance constraints to ensure that the predicted near future demand is served with a certain probability. Schuijbroek, Hampshire, and Van Hoeve (2017) develop a scalable approximate solution by clustering the base stations using maximum spanning star (MAXSPS) and allocate one vehicle in each cluster so as to meet the service level requirements. Furthermore, they represent the problem as a clustered vehicle routing problem [CluVRP] (Battarra, Erdogan, and Vigo 2014) and solve it in an online fashion. Contardo, Morency, and Rousseau (2012) present a scalable myopic repositioning solution by considering the current demand that was recently observed and solve it using Dantzig-Wolfe and Benders decomposition techniques. Recently, Lowalekar et al. (2017) propose a scalable online repositioning solution using multistage stochastic optimization and online anticipatory algorithms. Ghosh, Trick, and Varakantham (2016) propose a robust and online repositioning approach for the vehicles to counter the uncertainty in future demand. In contrast to the rolling horizon based online solutions, Shu et al. (2010; 2013) consider the future expected demand for a long period to deal with the future demand surges and propose an optimization model for dynamic repositioning to minimize the number of unsatisfied customers. However, they did not consider the specific routing constraints and the physical limitations of the vehicles in their model. Ghosh et al. (2015; 2017) overcome this concern by jointly considering the dynamic repositioning of bikes in conjunction with the routing problem for vehicles. Our approach differs from this thread of research as we are utilizing the bike trailers for repositioning and crowdsourcing the tasks to customers in contrast to using vehicles with dedicated staffs for repositioning.

The last thread of research focuses on incentivizing customers and utilizing trailers for rebalancing the system. There has been existing research in bike sharing (Singla et al. 2015; Pfrommer et al. 2014) and car sharing (Chow and Yu 2015; Mareček, Shorten, and Yu 2016) that present pricing mechanisms to provide incentives to users for assisting with repositioning. However, this line of work has taken a myopic (individual station) view on whether a bike or car is required at a station. Furthermore, unlike car sharing (Chow and Yu 2015), the BSS operators cannot order users based on their utility and operate within tight budget constraints. In this work, we provide an end to end system that takes the global view (all stations) of the repositioning requirements and incentives their execution within the budget constraints.

On the other hand, O'Mahony and Shmoys (2015) predict the service level requirements for base stations in rush hours and introduce the notion of repositioning with bike trailers, by matching each trailer to its suitable producer and consumer stations, based on the assessment of inventory state of the base stations. However, they assume that all the tasks for the trailers can be achieved with dedicated staff which is not an economically viable option. In contrast, we propose an optimization model to generate the repositioning tasks for trailers and design a mechanism to crowdsource those tasks to the users while ensuring the given budget constraints.

Model: DRRPT

In this section, we formally describe the generic model of Dynamic Repositioning and Routing Problem using Trailers (DRRPT) extending from the DRRP model introduced by Ghosh et al. (2015; 2017). DRRPT is compactly represented using the following tuple:

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

$\mathcal{S}$ denotes the set of base stations where $C_s^{\#}$ represents the capacity of the station $s \in \mathcal{S}$. We have a set of bike trailers $\mathcal{V}$ where C_v^* denotes the number of bike slots in the trailer $v \in \mathcal{V}$. $d^{\#,0}$ represents the initial distribution of bikes in the stations. σ_v^0 symbolises the initial locations of the trailers, i.e., $\sigma_v^0(s)$ is fixed to 1 if trailer v is stationed at s initially and 0 otherwise. $\boldsymbol{H}$ denotes a two dimensional matrix that stores the relative distance between each pair of stations. $\boldsymbol{F}$ represents a set of K discrete training demand scenarios. Specifically, $F_{s,s'}^k$ denotes the demand for the planning period for scenario k that arises at station s and reaches station s' in the next time step. Finally, B denotes the amount of budget allocated for the repositioning tasks for a given planning period.

We make the following assumptions for the ease of explanation and representation. However, these assumptions can easily be relaxed with minor modifications to our methods; (a) In the similar direction of Ghosh, Trick, and Varakantham (2016), we assume that users who carry bikes and trailers at decision epoch t always return their bikes at the beginning of the decision epoch t + 1; (b) Customers are impatient in nature and leave the system if they encounter an empty station. On the other hand, they return their bikes to the nearest available station if the destination station is full.

Solving DRRPT

We propose a rolling horizon framework for solving DRRPT, where the following two components are run continuously at each time step:

- Generate repositioning tasks for the next time step;
- Mechanism to incentivize execution of tasks (within the central budget constraints) by interested users.

Generating Repositioning Tasks

In this section, we describe the method for computing repositioning tasks for the trailers and also estimate the valuations of those tasks from center's perspective. As a trailer can travel at most to one station in each time step (equivalent to bikes), the repositioning task for a trailer is to pickup a certain number of bikes from the neighbourhood of its origin station and drop them to another station. To formally represent the repositioning tasks, we introduce the following decision variables:

- $y_{s,v}^+$ denotes the number of picked up bikes by trailer v from station s;
- $y_{s,v}^-$ denotes the number of bikes dropped off by trailer v at station s;
- $b_{s,v}^+$ is a binary decision variable which is set to 1 if trailer v picks up bikes from station s and 0 otherwise;
- $b_{s,v}^-$ represents a binary decision variable which is set to 1 if trailer v returns bikes at station s and 0 otherwise.

$$\begin{aligned}
&\min_{\boldsymbol{y}} \; \sum_{s,k} L_s^k && (1)\\
&\textbf{s.t. } L_s^k \geq \sum_{s'} F_{s,s'}^k - \big(d_s^{\#,t} + \sum_v (y_{s,v}^- - y_{s,v}^+)\big), \forall k, s && (2)\\
&y_{s,v}^+ \leq b_{s,v}^+ \cdot \min(d_s^{\#,t}, C_v^*), \qquad \forall s, v && (3)\\
&\sum_v y_{s,v}^+ \leq d_s^{\#,t}, \qquad \forall s && (4)\\
&\sum_v y_{s,v}^- \leq C_s^{\#} - d_s^{\#,t}, \qquad \forall s && (5)\\
&y_{s,v}^- = b_{s,v}^- \cdot \sum_s y_{s,v}^+, \qquad \forall s, v && (6)\\
&(b_{s,v}^+ + b_{s',v}^- - 1) \cdot H_{s,s'} \leq H_{max}, \qquad \forall s, s', v && (7)\\
&\sum_s b_{s,v}^+ = 1, \qquad \forall v && (8)\\
&\sum_{s \notin G_v} b_{s,v}^+ = 0, \qquad \forall v && (9)\\
&\sum_s b_{s,v}^- = 1, \qquad \forall v && (10)\\
&b_{s,v}^+, b_{s,v}^- \in \{0,1\}; y_{s,v}^+, y_{s,v}^- \leq C_v^*; L_s^k \geq 0 && (11)
\end{aligned}$$

Table 1: TASKGENERATION(**F**,t,$\boldsymbol{d^{\#}}$,drrpt)

In addition, we use the symbol G_v to denote the set of neighbouring stations from where vehicle v is allowed to pick up bikes. A station is included in G_v if it is situated within a threshold distance from the origin station of trailer v. Our goal is to compute the best routing and repositioning strategy for each of the bike trailers so as to minimize the total number of expected lost demand over K training demand scenarios. Let L_s^k denotes the lost demand at station s for demand scenario k, after the repositioning tasks are achieved. We represent the problem of minimizing expected lost demand as a Mixed Integer Linear Programme (MILP). The MILP for the task generation is compactly represented in Table (1). Our objective (delineated in expression 1) is to minimize the expected lost demand (equivalent to total lost demand, as each scenario has equal probability) over all the K training scenarios. The constraints associated with this repositioning task generation are described as follows:

1. **Compute the lost demand as the deficiency in supply of bikes:** The number of bikes present in a station s after accomplishing the repositioning task is estimated as $d_s^{\#,t} + \sum_v (y_{s,v}^- - y_{s,v}^+)$. Therefore, constraints (2) ensure that the number of lost demand at station s for scenario k is lower bounded by the difference between demand and supply of bikes at that station. Note that, as we are minimizing the sum of lost demand over all the scenarios, these constraints are sufficient alone to compute the exact number of loss in customer demand.
2. **Trailer capacity is not exceeded while picking up bikes:** Constraints (3) ensure that the number of bikes picked up by trailer v from station s is bounded by the minimum value between the number of bikes present in the station and the capacity of the trailer.

3. **Total number of bikes picked up from a station is less than the available bikes:** As multiple trailers can pick up bikes from the same station, constraints (4) enforce that the total number of picked up bikes by all the trailers from station s during the planning period t is bounded by the number of bikes present in the station, $d_s^{\#,t}$.
4. **Station capacity is not exceeded while dropping off bikes:** Constraints (5) ensure that the total number of dropped off bikes at station s is bounded by the number of available slots for bikes at that station.
5. **A trailer should return the exact number of bikes it has picked up:** Note that $b^-_{s,v}$ is the binary decision variable that controls the drop-off location for the trailer v. Therefore, constraints (6) enforce that the number of bikes dropped off by a trailer in a station is exactly equals to the number of picked up bikes if the station is visited.
6. **Total traveling distance for a trailer is bounded by a threshold value:** To represent the physical limitation of route, we need to ensure that the total distance travelled by a trailer in a given planning period is within a few kilometers. Constraints (7) enforce this condition by ensuring that the distance between the pick up and the drop-off station for a trailer is bounded by a threshold value, H_{max}.
7. **A trailer should pick up bikes from one station only:** Constraints (8) enforce this condition by allowing only one pick up decision variable to be set to 1 for each trailer.
8. **The pick up location for a trailer should be within the geographical proximity of its origin station:** Constraints (9) assure this requirement by fixing all the pick up decision variables for trailer v to 0 for the stations which does not belong to its nearby station set, G_v.
9. **A trailer should return bikes to one station only:** Constraints (10) ensure this condition by allowing only one drop-off decision variable to be set to 1 for each trailer.

Note that, constraints (6) are non-linear in nature. However, one component in the right hand side is a binary variable. Therefore, we can easily linearize them using the following constraints (12)-(14).

$$y^-_{s,v} \leq C^*_v \cdot b^-_{s,v} \qquad \forall s, v \quad (12)$$

$$y^-_{s,v} \leq \sum_s y^+_{s,v} \qquad \forall s, v \quad (13)$$

$$y^-_{s,v} \geq \sum_s y^+_{s,v} - (1 - b^-_{s,v}) \cdot C^*_v \qquad \forall s, v \quad (14)$$

Although we are using big-M method for the linearization, the upper bound for the pickup or drop-off variable (or alternatively the value of M) is the capacity of the trailer which is relatively small and therefore, these constraints are computationally inexpensive.

Mechanism to Incentivize Task Execution within Budget Constraints

Once we determine the tasks, our goal is to design a mechanism which allocates the tasks among the users who are interested in executing these tasks and generate a payment method to ensure that the users bid for the tasks truthfully. If the payment method does not ensure truthful behaviour, then either the bike sharing operators are unhappy (as they pay more money to users than required) or users are unhappy (as they get paid less) while repositioning bikes through trailers.

To design a mechanism for crowdsourcing the repositioning tasks, the first step is to compute the value of the tasks from center's perspective. As our goal is to minimize the expected lost demand, the valuation of the task is proportional to the expected lost demand reduced by the trailer job over all the training demand scenarios. Specifically, the value of task for trailer v is defined as follows (ξ represents unit value of lost demand to compute overall value):

$$U(v) = \frac{\xi}{K} \sum_{k,s} \Big[\min\big(\max(\sum_{s'} F^k_{s,s'} - d_s^{\#,t}, 0), y^+_{s,v} \big) - \min\Big(\max\big(y^-_{s,v} - (d_s^{\#,t} - \sum_{s'} F^k_{s,s'}), 0 \big), y^-_{s,v} \Big) \Big] \quad (15)$$

Intuitively this value is the weighted difference in reduced lost demand using the trailer minus increase in lost demand due to moving bikes using trailer. The first term in equation (15) computes the expected lost demand reduced by trailer v in its destination station over K scenarios. The second term computes the expected lost demand arising because of the pickup decision by trailer v at its origin station.

We assume that the set of interested users for each pair of tasks are disjoint. One user can execute a single task in any given decision epoch, so this assumption can be easily enforced. To ensure this assumption is satisfied, we can either associate a huge penalty for bidding on multiple tasks or discard all bids of a user except the first one. Once all the bids arrive, the goal of the center is two-fold: (a) Design an incentive compatible mechanism to ensure that users bid truthfully on every task; (b) Allocate the tasks in a fashion that maximizes the efficiency of the entire system while satisfying the budget feasibility.

Observation 1 *As the set of bidders for different tasks are pairwise disjoint and the mechanism initiates once all the bid information is available, the tasks are primarily independent but coupled by the central budget constraint. Therefore, the mechanism or payment method can be designed for each of the tasks separately. However, the final allocation of tasks should be accomplished in a fashion so that the budget feasibility is ensured.*

By exploiting observation (1), we design a mechanism for each of the tasks separately. Let the set of repositioning tasks be $\mathcal{T} = \{1, ..., |\mathcal{V}|\}$. We begin the discussion with the mechanism design for a single task for trailer v. Let, $N_v = \{1, ..., n_v\}$ represents the set of rational users who are bidding privately to the center for the task of trailer v. Each user $i \in N_v$ privately reveals their type $\theta_i =< C_i(v) >$, where $C_i(v)$ denotes their private cost for executing the task of trailer v. The center's profit for the bid of user i is defined as $W_i(v) = U(v) - C_i(v)$. We reject a bid from a user i if $C_i(v) > U(v)$, which ensures that $W_i(v)$ is always positive. Our goal is to assign the task to a bidder so that the center's profit is maximized and design a payment method to ensure that users always bid truthfully. We use the standard Vickrey-Clarke-Groves [VCG] mechanism (Vickrey 1961; Clarke 1971; Groves 1973) to solve this problem.

According to this mechanism, the task is always allocated to the lowest bidder, but the lowest bidder gets paid the bid of the second lowest bidder. For instance, if there are 3 bids of 10$, 12$ and 14$ to perform a repositioning task, then the task is allocated to bid 1 and the person putting in bid 1 gets paid 12$. More formally, let $\lambda^* = \{0,1\}^{N_v}$ denotes the allocation of the task so that the center's profit is maximized.

$$\lambda_i^*(v) = \left\{ \begin{array}{ll} 1 & \textbf{if } i = \text{argmax}_{j \in N_v} W_j(v) \\ 0 & \textbf{Otherwise} \end{array} \right\}$$

Then the payment to the user i for task v is computed using the following expression:

$$P_i(v) = \lambda_i^*(v)\big[U(v) - \max_{j \neq i} W_j(v)\big] = \lambda_i^*(v)\big[\min_{j \neq i} C_j(v)\big] \tag{16}$$

Equation (16) indicates that the payment for user i is the second lowest cost revealed in the bid process if the task is allocated to him, otherwise the payment is set to 0.

Since, we directly adapt the standard VCG mechanism, the mechanism for single task is truthful or incentive compatible. However, this does not ensure incentive compatibility over all tasks, as there is a budget constraint. We now provide a method that will ensure incentive compatibility over all tasks without violating the budget feasibility.

Ensuring the Budget Feasibility: As mentioned previously, the BSS operators work within a fix budget B. We have a set of tasks $\mathcal{T} = \{1, ..., |\mathcal{V}|\}$, where each task $v \in \mathcal{T}$ has a valuation, $U(v)$ (computed using equation 15) and the payment for the task is denoted by $P(v)$ (computed using equation 16). Our goal is to allocate the tasks in a fashion that maximizes the overall valuation of the center while the total payment is bounded by the given budget, B. Let $x(v)$ denotes a binary decision variable which is set to 1 if task v is allocated and 0 otherwise. We compactly represent the problem as an Integer Program (IP) in table (2).

$$\max_x \sum_{v \in \mathcal{T}} x(v) \cdot U(v) \tag{17}$$

$$\text{s.t.} \sum_{v \in \mathcal{T}} x(v) \cdot P(v) \leq B \tag{18}$$

$$x(v) \in \{0,1\} \qquad \forall v \in \mathcal{T} \tag{19}$$

Table 2: TASKALLOCATIONIP($\boldsymbol{U}, \boldsymbol{P}, \mathcal{T}, B$)

Our objective in expression (17) aims to find an allocation of the tasks so that the cumulative valuation from the center's perspective is maximized. Constrains (18) enforce that the total payment made to the users due to the resulting allocation should respect the given budget B. The IP in Table (2) is exactly equivalent to the 0/1 knapsack problem which is a known NP-Hard problem. However, we can employ the well-known dynamic programming (DP) approach (refer to chapter 6 of Dasgupta, Papadimitriou, and Vazirani, 2006) to speed up the solution process. The complexity of such a DP approach is $O(|\mathcal{T}| \cdot B)$ in comparison to the brute force method that has the complexity of $O(2^{|\mathcal{T}|})$.

Proposition 1 *The mechanism for task allocation for the trailers in bike sharing system is incentive compatible (IC), individually rational (IR) and economically efficient (EE).*

Proof: The mechanism for single task satisfies the IC and IR property as it follows the standard VCG mechanism. As all the tasks are independent and payments are made for a subset of tasks to ensure the budget feasibility, all the allocated tasks satisfy the IC and IR property. Hence, the budget feasible mechanism for the entire BSS meets the requirements to satisfy the IC and IR property. Finally, the mechanism maximizes the difference between center's valuation and the cost for executing the task which is equivalent to expected total welfare, hence, our mechanism satisfies the EE property. ■

Overall Flow of Our Approach

To better understand the overall flow of our approach, we provide Algorithm (1). We begin by solving the MILP of Table (1) to generate the repositioning tasks for each of the trailer to better satisfy customer demand over a set of training demand scenarios. Then the values of the tasks from center's perspective are computed using equation (15) and broadcasted to the users. Next, a set of rational users bid for the tasks privately. Once all the bids are submitted, we employ the standard VCG mechanism to generate the payment (refer to equation 16) for each task. Finally, we allocate budget to tasks (and make payments only if the task can be allocated money) by solving a 0/1 knapsack problem that maximizes the global welfare of the system without violating the budget constraints of the operator.

```
Algorithm 1: solveRepositioning(drrpt, $t$, $\boldsymbol{d}^{\#}$, $\boldsymbol{F}^{t}$, $B$)
Initialize: $\boldsymbol{Y}^+, \boldsymbol{Y}^- \leftarrow 0$ ;
$\boldsymbol{Y}^+, \boldsymbol{Y}^- \leftarrow$ TASKGENERATION($\boldsymbol{F}^{t}, t, \boldsymbol{d}^{\#}$, drrpt);
for each $v \in \mathcal{V}$ do
    $U(v) \leftarrow$ COMPUTETASKVALUE($Y_v^+, Y_v^-$);
for each $v \in \mathcal{V}$ do
    $\boldsymbol{C}(v) \leftarrow$ COLLECTBIDS($Y_v^+, Y_v^-, U(v)$);
for each $v \in \mathcal{V}$ do
    $P(v) \leftarrow$ GENERATEPAYMENT($U(v), \boldsymbol{C}(v)$);
$\boldsymbol{X} \leftarrow$ TASKALLOCATION($\boldsymbol{U}, \boldsymbol{P}, \mathcal{V}, B$);
for each $v \in \mathcal{V}$ do
    $\boldsymbol{Y}_v^+ \leftarrow \boldsymbol{Y}_v^+ \cdot X(v)$;
    $\boldsymbol{Y}_v^- \leftarrow \boldsymbol{Y}_v^- \cdot X(v)$;
return $\boldsymbol{Y}^+, \boldsymbol{Y}^-$;
```

Empirical Evaluation

In this section, we explain the simulation model used to execute the tasks, the benchmark approaches implemented for the computational comparisons and the experimental results.

Simulation Model

Once the repositioning tasks for the trailers and their allocations are determined for a time step, we execute them on a simulator (adapted from Ghosh, Trick, and Varakantham 2016) for evaluating their performance on the realized demand scenario for that particular time step.

Let, $f^t_{s,s'}$ denotes the number of customers who arrive in station s at time step t and plan to reach station s' at the beginning of time step $t+1$. Let, $d^{\#,t}_s$ represents the number of bikes present in station s at time step t after the repositioning tasks are completed. Due to the repositioning, the number of bikes available in stations changes and therefore, the flows of bikes by the customers also change in comparison to the observed data denoted by $\boldsymbol{f}$. However, a reasonable assumption employed in previous works (Shu et al. 2013; Ghosh et al. 2017; Ghosh, Trick, and Varakantham 2016) for any configuration is that the aggregated transition probabilities between stations that is observed in the data remain the same during execution.

Therefore, the flows of bikes between the stations are determined based on the following two cases: (a) If the arrival demand at a station is less than the number of bikes present in that station, then all the customers are able to hire bikes; (b) If the arrival demand at a station is higher than the number of bikes present in that station, then the actual flow of bikes (denoted as $x^t_{s,s'}$) is computed using the relative ratio $\frac{f^t_{s,s'}}{\sum_{s'} f^t_{s,s'}}$ as shown in equation (20).

$$x^t_{s,s'} = \left\{ \begin{array}{ll} f^t_{s,s'} & \textbf{if } \sum_{s'} f^t_{s,s'} \leq d^{\#,t}_s \\ \frac{f^t_{s,s'}}{\sum_{\tilde{s}} f^t_{s,\tilde{s}}} \cdot d^{\#,t}_s & \textbf{Otherwise} \end{array} \right\} \quad (20)$$

Once the actual flow of bikes by the customers at time step t is determined, we calculate the distribution of bikes in station s at time step $t+1$ as the sum of un-hired bikes at time step t, the net incoming bikes in station s at the beginning of time step $t+1$ and the net drop-off bikes at station s by the trailers at time step $t+1$ (i.e., $Y^{-,t+1}_s - Y^{+,t+1}_s$).

$$d^{\#,t+1}_s = d^{\#,t}_s + \Big[\sum_{\tilde{s}} x^t_{\tilde{s},s} - \sum_{s'} x^t_{s,s'}\Big] + \Big[Y^{-,t+1}_s - Y^{+,t+1}_s\Big] \quad (21)$$

Equation (21) for computing the number of bikes in station s at time step $t+1$ (i.e., $d^{\#,t+1}_s$) does not take into account the station capacity constraints. To handle such boundary conditions and to ensure the capacity constraints are considered for the stations, we transfer extra bikes (i.e., $d^{\#,t+1}_s - C^{\#}_s$) to the nearest available station if $d^{\#,t+1}_s$ exceeds the station capacity, $C^{\#}_s$. In our experimental results, we show these extra numbers as the lost demand at the time of return. Once the distribution of bikes across the stations for time step $t+1$ is obtained, we utilize this information to compute the repositioning strategy for trailers for time step $t+1$. This iterative process continues until we reach the last decision epoch.

Experimental Setup

We conducted our experiments on a real-world data set[1] of *Hubway* BSS. The *Hubway* data set contains the following details: (1) Customer trip records, from which we compute demand for each origin destination pair at each time step; (2) The number of stations, their capacity and initial distribution of bikes at the stations; (3) Geographical locations of base stations to calculate the distance between two stations; (4) The number of vehicles used for repositioning and their capacity. Furthermore, we generate two sets of synthetic demand scenarios using *Poisson* distribution with the mean computed from real-world data set. More specific details about these synthetic data sets are mentioned later. We evaluate our approach with respect to the key performance metric of loss in customer demand. We compare the utility of our approach with two existing benchmark approaches.

Benchmark-1: Static Repositioning implies the practice of no repositioning during the day. The vehicles are only used at the end of the day to rebalance stations to achieve a predefined inventory level. We use this as a baseline approach where no repositioning is done during the planning period.

Benchmark-2: Dynamic Repositioning implies the practice of repositioning idle bikes using vehicles during the day to meet the expected future demand. We adapted a recently developed scenario based approach from Ghosh, Trick, and Varakantham (2016). In their iterative approach, a worse demand scenario is generated in each iteration to counter the repositioning strategy of the current iteration and then they produce a robust repositioning solution by considering all the previously generated scenarios. However, for a fair comparison with our approach (as shown in Table 1), we make the following modifications in their optimization model: (1) We take the exact set of training demand scenarios used in our approach rather than generating the worse case scenarios; (2) We minimize the total number of lost demand over all the demand scenarios (equivalent to our objective function of Table 1) in contrast to optimizing for the worst case.

To ensure a fair comparison, all the benchmark approaches and our approach are evaluated on the simulation model described in the previous subsection.

Empirical Results

We now show the performance[2] of our approach on *Hubway* data set. The *Hubway* BSS consists of 95 base stations and 3 vehicles. We study with 10 trailers and their capacity is assumed to be three in our default settings of experiments. We take 6 hours of planning horizon in the morning peak (6AM-12PM) and the duration of each decision epoch is considered as 30 minutes. The demand scenarios are generated from three months of historical trip data. As the trip data only contains successful bookings and does not capture the unobserved lost demand, we employ a micro-simulation model (courtesy: Ghosh, Trick, and Varakantham 2016) with 1 minute of time step to determine the time slots when a station was empty and introduce artificial demand at the empty station based on the observed demand at that station in previous time step. We demonstrate three sets of results on the *Hubway* data set:

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

[2] All the linear optimization models were solved using IBM ILOG CPLEX Optimization Studio V12.5. incorporated within python code.

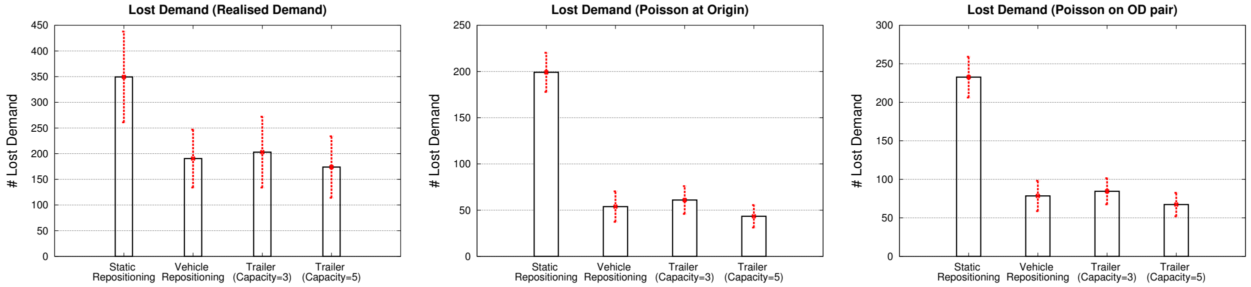

Figure 1: Lost demand statistics for (a) Demand scenarios from real-world data; (b) Demand scenarios follow *Poisson* distribution at origin station; (c) Demand scenarios follow *Poisson* distribution for each OD pair.

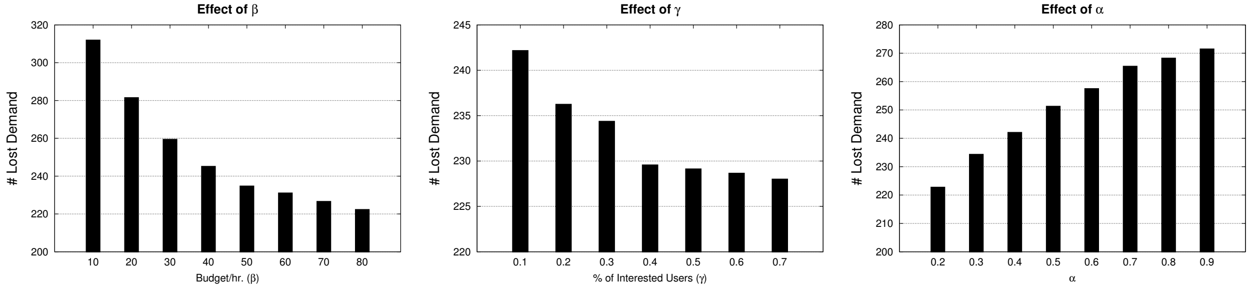

Figure 2: (Average) Lost demand statistics for varying (a) Allocated budget ($\boldsymbol{\beta}$) [$\alpha = 0.3, \gamma = 0.3$]; (b) Percentage of users interested in trailer tasks ($\boldsymbol{\gamma}$) [$\alpha = 0.3, \beta = 50$]; (c) Ratio of lower and upper bound of bids ($\boldsymbol{\alpha}$) [$\beta = 50, \gamma = 0.3$].

- The performance comparison between our approach and the benchmarkss in terms of reducing the lost demand;
- The effect of key tunable input parameters on the mechanism design over a wide range of demand scenarios;
- Runtime performance of our approach.

Performance comparison: To evaluate the performance of our approach, we produce three types of demand scenarios: (1) We took the real demand data for 60 weekdays. The actual demand is estimated by introducing artificial demand at empty stations using a similar heuristic as discussed earlier. We use 20 days of demand scenarios for training purpose and other 40 days of demand for testing; (2) We generate 100 demand scenarios, where the arrival demand at each station is generated using *Poisson* distribution with the mean computed from historical data. Similar to Shu et al. (2013), we assume that customers reach their destination station with a fixed probability; (3) We generate 100 demand scenarios, where demand for each origin destination [OD] pair at each time step is computed using *Poisson* distribution. For the demand scenario types 2 and 3, we use 30 demand scenarios for training and 70 demand scenarios for testing.

For all the three types of demand scenarios, we compute the cumulative lost demand at the time of bike pickup and return for the following four approaches: (a) Static repositioning, i.e., no rebalancing is done during the planning period; (b) Dynamic repositioning using 3 existing vehicles; (c) Dynamic repositioning using 10 trailers, each having a capacity of 3; (d) Dynamic repositioning using 10 trailers, each having a capacity of 5. For this set of experiments, we assume that there is sufficient budget available to allocate all the tasks. Therefore, we directly took the repositioning solution from Table (1) for evaluation. Figure (1) depicts the average number of lost demand along with standard deviation for all the three types of demand scenarios. Figure 1(a) shows the lost demand statistics on the real-world demand scenarios. By utilizing trailers with capacity 3, the average lost demand over 40 testing scenarios reduces by 41% over the no repositioning approach. The repositioning solutions for the trailers with capacity 3 are also proven to be highly competitive to the solutions achieved by vehicles. As expected, the repositioning solutions for the trailers with capacity 5 produce better results and outperform the lost demand obtained by 3 carrier vehicles. Similar performance statistics are shown in Figures 1(b) and 1(c) for the demand scenarios generated using *Poisson* distribution at origin station and for each OD pair respectively. In both the settings, we observe a consistent pattern that the repositioning solution using trailers with capacity 3 reduces the average lost demand over 70 test scenarios by 69% and 63% in comparison to the baseline approach. Moreover, both the figures clearly demonstrate that the solutions for trailers with capacity 5 are better than the fuel burning mode of repositioning by the vehicles.

Effect of tunable parameters: In the next set of results we demonstrate the performance of our approach by varying the different input parameters of the mechanism. We employ the real-world demand scenarios (demand scenario type 1) for these experiments, where 20 demand scenarios are used for training and the evaluation is done on other 40 scenarios. The outcome of the mechanism depends on the following three input parameters:

- Hourly budget allocated by the operators (β): Ideally the BSS operators allocate a fixed amount beforehand for the repositioning tasks. In our default settings of experiments we have fixed the hourly budget to 50 dollars[3];

- Percentage of users interested in trailer tasks (γ): To execute a mechanism, it is important to compute the number of users bidding for each trailer task. Typically, a certain percentage of users whose origin and destination location is within Δ kilometer of the pickup and drop-off location of the trailer, are the potential users interested in executing the task and bid for it. In our experiments we set the value of Δ to 1 kilometer[4]. We use the default value of γ as 30%;
- Ratio of lower and upper bound of bids (α): The third and most important parameter for the mechanism is the bid values submitted privately by the users. We generate the bid values using *Gaussian* distribution[5] from the range $[C_{\min}, C_{\max}]$. As the upper limit of the bid value for task v is bounded by its valuation $U(v)$, we set the $C_{\max}$ for the task of trailer v to $U(v)$. The value of $C_{\min}$ is set to $\alpha C_{\max}$. As the bids are generated from a distribution, we run the mechanism 100 times for every task and use the expectation over 100 runs as the payment. The default value of α is set to 30%.

Figure (2) depicts the effect of the tunable parameters on the performance of our approach. Figure 2(a) plots the average lost demand over 40 days of testing demand scenarios, where we vary the hourly budget (β) in the X-axis from 10 dollars to 80 dollars. As expected, the average number of lost demand reduces monotonically as we increase the hourly budget. Due to the randomness in bid values in different runs, the lost demand might increase for some scenarios, even after increasing the hourly budget. We observe that for more than 78% of the cases, lost demand decreases if we increase the hourly budget by 10 dollars.

Figure 2(b) plots the average lost demand over 40 testing demand scenarios, when we vary the interest rate of the users (γ) in the X-axis from 10% to 70%. The average number of lost demand reduces monotonically as we increase the interest rate of users, because increasing the interest rate implies that additional bids are submitted to the center and therefore, the likelihood of the payment value reduces which in turn enable us to execute extra trailer tasks within the given budget, hence, the number of expected lost demand reduces. We observe that the lost demand decreases for around 60% of the cases, if we increase γ by 10%.

Figure 2(c) plots the average lost demand over 40 testing demand scenarios, where we vary the ratio of the lower and upper bound of the bids (α) in the X-axis from 20% to 90%. Increasing the value of α indicates that the lower bound of the bids increases, so the expected bid value also increases. Now, increase in the bid values implies that the expected payment for the tasks also increases and the number of tasks that can be executed within a fixed budget decreases, hence, the number of expected lost demand also increases. As expected, Figure 2(c) clearly depicts that the average number of lost demand increases monotonically as we increase the value of α. For around 74% of the cases, the lost demand increases if we increase α by 10%.

Runtime performance: In the last set of results, we show the runtime performance of our approach in comparison to the repositioning solution of the vehicles on the real-world demand scenarios. The time to find a repositioning solution is a crucial factor in our settings, as we are generating the strategy after every 30 minutes of interval. Figure (3) depicts the runtime performance where in the X-axis we vary the number of decision epochs and the Y-axis denotes the cumulative runtime in logarithmic scale. For every value of decision epoch, our approach was able to solve the problem within a couple of seconds with 95 stations and 20 training scenarios. On the other hand, it took more than 15 minutes for each decision epoch to generate the solutions for the vehicles with the same number of training scenarios.

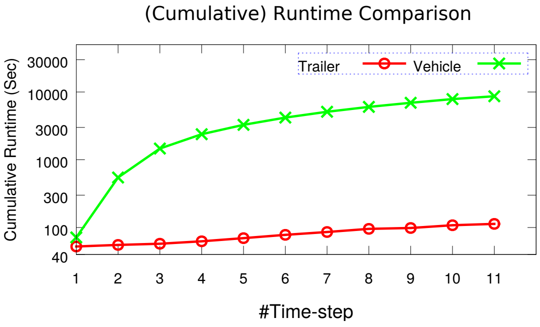

Figure 3: (Cumulative) Runtime comparison between the repositioning solutions of vehicles and trailers.

Conclusion

In this paper we explore the dynamic repositioning problem in bike sharing systems with the help of bike trailers. We propose a novel optimization model to generate the repositioning tasks for the trailers to better meet the customer demand. Additionally, we design a budget feasible incentive compatible (incentivizes truth telling) mechanism to crowdsource the tasks among the users who are interested in executing those tasks. The empirical results on a real-world data set show that our green mode of repositioning is economically viable and highly efficient in terms of reducing the lost demand. In future this work can be extended in the following directions: (a) Developing a budget feasible mechanism by considering the uncertainties in completion time of the trailer tasks; (b) Developing a model that jointly consider the dynamic repositioning problem for vehicles and trailers and discover an efficient solution while ensuring the central budget constraint.

[3] The link: http://www.bls.gov/oes/current/oes533033.htm from the US Department of Labor provides hourly salaries for drivers operating light trucks that are used for repositioning the bikes. It shows that the median hourly cost for a hired driver is 14 dollars. Therefore, the hourly budget for 3 existing vehicles including the fuel cost for routing would be around 50 dollars.

[4] We experimented with Δ as 0.5 kilometer and observe similar results as shown in Figure 2.

[5] We choose *Gaussian* bids, because existing work (Singla et al. 2015) in bike sharing that experimented with real humans employed *Gaussian* distribution for representing user bids.

References

Battarra, M.; Erdogan, G.; and Vigo, D. 2014. Exact algorithms for the clustered vehicle routing problem. *Operations Research* 62(1):58–71.

Benchimol, M.; Benchimol, P.; Chappert, B.; De La Taille, A.; Laroche, F.; Meunier, F.; and Robinet, L. 2011. Balancing the stations of a self service “bike hire” system. *RAIRO-Operations Research* 45(1):37–61.

Chalasani, P., and Motwani, R. 1999. Approximating capacitated routing and delivery problems. *SIAM Journal on Computing* 28(6):2133–2149.

Chemla, D.; Meunier, F.; and Wolfler Calvo, R. 2013. Bike sharing systems: Solving the static rebalancing problem. *Discrete Optimization* 10(2):120–146.

Chow, Y., and Yu, J. Y. 2015. Real-time bidding based vehicle sharing. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, 1829–1830. International Foundation for Autonomous Agents and Multiagent Systems.

Clarke, E. H. 1971. Multipart pricing of public goods. *Public choice* 11(1):17–33.

Contardo, C.; Morency, C.; and Rousseau, L.-M. 2012. *Balancing a dynamic public bike-sharing system*, volume 4. CIRRELT.

Dasgupta, S.; Papadimitriou, C. H.; and Vazirani, U. V. 2006. *Algorithms*. McGraw-Hill, Inc., New York, NY.

Di Gaspero, L.; Rendl, A.; and Urli, T. 2013. A hybrid ACO+ CP for balancing bicycle sharing systems. In *Hybrid Metaheuristics*. Springer. 198–212.

Di Gaspero, L.; Rendl, A.; and Urli, T. 2015. Balancing bike sharing systems with constraint programming. *Constraints* 1–31.

Ghosh, S.; Varakantham, P.; Adulyasak, Y.; and Jaillet, P. 2015. Dynamic redeployment to counter congestion or starvation in vehicle sharing systems. In *International Conference on Automated Planning and Scheduling (ICAPS)*.

Ghosh, S.; Varakantham, P.; Adulyasak, Y.; and Jaillet, P. 2017. Dynamic repositioning to reduce lost demand in bike sharing systems. *Journal of Artificial Intelligence Research* 58:387–430.

Ghosh, S.; Trick, M.; and Varakantham, P. 2016. Robust repositioning to counter unpredictable demand in bike sharing systems. In *International Joint Conference on Artificial Intelligence (IJCAI)*.

Groves, T. 1973. Incentives in teams. *Econometrica: Journal of the Econometric Society* 617–631.

Lowalekar, M.; Varakantham, P.; Ghosh, S.; Jena, S. D.; and Jaillet, P. 2017. Online repositioning in bike sharing systems. In *International Conference on Automated Planning and Scheduling (ICAPS)*.

Mareček, J.; Shorten, R.; and Yu, J. Y. 2016. Pricing vehicle sharing with proximity information. In *Big Data and Smart City (ICBDSC), 2016 3rd MEC International Conference on*, 1–7. IEEE.

Meddin, R., and DeMaio, P. 2016. The bike-sharing world map. In *http://www.bikesharingworld.com/*.

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

Nair, R.; Miller-Hooks, E.; Hampshire, R. C.; and Bušić, A. 2013. Large-scale vehicle sharing systems: Analysis of Vélib'. *International Journal of Sustainable Transportation* 7(1):85–106.

O'Mahony, E., and Shmoys, D. B. 2015. Data analysis and optimization for (Citi) bike sharing. In *AAAI Conference on Artificial Intelligence (AAAI)*.

Pfrommer, J.; Warrington, J.; Schildbach, G.; and Morari, M. 2014. Dynamic vehicle redistribution and online price incentives in shared mobility systems. *IEEE Transactions on Intelligent Transportation Systems* 15(4):1567–1578.

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

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

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

Shu, J.; Chou, M.; Liu, Q.; Teo, C.-P.; and Wang, I.-L. 2010. Bicycle-sharing system: deployment, utilization and the value of re-distribution. In *National University of Singapore-NUS Business School, Singapore*.

Shu, J.; Chou, M. C.; Liu, Q.; Teo, C.-P.; and Wang, I.-L. 2013. Models for effective deployment and redistribution of bicycles within public bicycle-sharing systems. *Operations Research* 61(6):1346–1359.

Singla, A.; Santoni, M.; Bartók, G.; Mukerji, P.; Meenen, M.; and Krause, A. 2015. Incentivizing users for balancing bike sharing systems. In *AAAI Conference on Artificial Intelligence (AAAI)*.

Vickrey, W. 1961. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of finance* 16(1):8–37.