Bike-Repositioning Using Volunteers: Crowd Sourcing with Choice Restriction

Jinjia Huang, Mabel C. Chou, Chung-Piaw Teo
Institute of Operations Research and Analytics, National University of Singapore, Singapore
oraahj@nus.edu.sg, bizchoum@nus.edu.sg, bizteocp@nus.edu.sg

Abstract

Motivated by the Bike Angels Program in New York's Citi Bike and Boston's Blue Bikes, we study the use of (registered) volunteers to re-position empty bikes for riders in a bike sharing system. We propose a method that can be used to deploy the volunteers in the system, based on the real time distribution of the bikes in different stations. To account for (random) route demand in the network, we solve a related transshipment network design model and construct a sparse structure to restrict the re-balancing activities of the volunteers (concentrating re-balancing activities on essential routes). We also develop a comprehensive simulation model using a threshold-based policy to deploy the volunteers in real time, to test the effect of choice restriction on volunteers (suitably deployed) to re-position bikes. We use the Hubway system in Boston (with 60 stations) to demonstrate that using a sparse structure to concentrate the re-balancing activities of the volunteers, instead of allowing all admissible flows in the system (as in current practice), can reduce the number of re-balancing moves by a huge amount, losing only a small proportion of demand satisfied.

Introduction

We study an innovative bike re-positioning scheme that helps maintain the availability of bikes for regular riders in the system. This is a volunteer-based re-balancing system adopted by many bike sharing systems (BSS) such as New York's Citi Bike Program - one of the largest BSS in the world. We are motivated by the following practical challenge - *When a volunteer arrives at a station, which destination do we want the volunteer to move an empty bike to (and be rewarded for making this move)?*

In May 2016, Citi Bike launched a pilot program, called Bike Angels, which pays registered members to redistribute the bikes themselves. Some other players, such as GoBike in San Francisco and Blue Bike in Boston have also started to adopt the Bike Angels system (Lefkowitz 2018). For a rigorous assessment of the impact of this program on bike re-balancing and dock relocation, see (Freund, Henderson, and Shmoys 2017; McGowen 2018; Freund et al. 2020). In the early days of the program, Citi Bike provided two static maps to the angels, one for the morning moves, and the other for the afternoon moves. They identified those stations in the system that typically faced shortage of bikes and docks during peak hours as drop-off stations and pick-up stations, respectively. Members are encouraged to earn points by taking a bike from a pick-up station and returning it to a neutral or drop-off station; or by taking a bike from a neutral station and returning it to a drop-off station. The neutral stations here function more like transit stations, where bikes can be brought in or out of the stations. We denote the above as the "fixed pick-up/drop-off structure." Note that the above essentially restricts the origin-destination (OD) pairs that the Angels can choose for re-balancing bikes - either between a pick-up or neutral station and a drop-off station, or between a pick-up and neutral station. The flows among other pairs of stations are not rewarded. Citi Bike has recently changed to a dynamic map, with the status of the stations determined according to real time usage information. This allows angels to re-balance bikes in any pair of stations depending on actual usage and forecast demands. As we will demonstrate later, while this natural deployment scheme gives volunteers the maximum flexibility to choose between pick-up/neutral and neutral/drop-off stations (i.e., fully flexible system), the system becomes less capable in directing the flow of empty bikes since the moves are decided by volunteers, which can lead to an excessive amount of redundant moves, without improving the number of rides satisfied in any significant manner.

Most of the earlier work in bike re-distribution focus on the operational problem of moving bikes using special vehicles deployed for this purpose. For related studies on the static bike re-positioning problem (SBRP), see (Benchimol et al. 2011; Angeloudis, Hu, and Bell 2014; Li et al. 2016; Kloimüllner and Raidl 2017; Schuijbroek, Hampshire, and Van Hoeve 2017). While these techniques are effective in reducing the re-positioning cost to some extent, the solutions could not incorporate the real-time demand of the users in their approach. To tackle this problem, other scholars study dynamic bike re-positioning problem (DBRP), which focus on assessing the demand and re-positioning bikes dynamically in search for better solution (Shu et al. 2013; Zhang et al. 2017; O'Mahony and Shmoys 2015; Ghosh et al. 2017). However, (Bonnotte et al. 2015) point out that bike re-positioning by vehicles could be costly and not effective. Different pricing schemes have been proposed to engage users to help reposition the bikes (Chemla et al. 2013; Pfrommer et al. 2014; Fricker and Gast 2016). (Singla et al. 2015) present a crowd-sourcing mechanism that engages the users using a smartphone application, and requests them to change their intended journeys in exchange for rewards. The incentive system is deployed in the bike sharing system: MVGmeinRad for a period of 30 days. Some researches argue that the operator could redistribute bikes by using the idle time of users who are willing to help, rather than by hiring staffs or inducing users to change their travel destinations (Aeschbach et al. 2015; Chung, Freund, and Shmoys 2018). In both operator-based and user-based BRP, the essential step in re-positioning bikes is to estimate and forecast which stations are critical stations in different time slots. The critical station refers to station that has no available bikes or docks for users to rent or return bikes. This is a well solved demand forecasting problem in machine learning, and has been actively studied in the Computer Science community (Liu et al. 2016; Cagliero et al. 2017). Different models for critical station detection have been proposed, for instance, regression-based models (Froehlich, Neumann, and Oliver 2008; Kaltenbrunner et al. 2010), classification models (Cagliero et al. 2017), statistical models (Alvarez-Valdes et al. 2016) and other engineering methods (Shahsavaripour 2015). Some recent research have focused on capturing demand uncertainty for better re-balancing decisions (Ghosh, Trick, and Varakantham 2016; Lu 2016; Jian, Hugo, and Lu 2019; Ghosh, Jing, and Patrick 2019).

In this paper, we use standard machine learning/econometric technique to determine the status of each station (as pick-up or drop-off station). We allow the volunteer to move a bike from a pick-up station to one of the adjacent drop-off stations (in the re-balancing network) to earn reward. Interestingly, we find that the set of OD pairs offered to the volunteers for re-positioning has a distinct impact on the efficiency of the re-balancing operations, and we propose a technique to construct the re-balancing network to ensure that most of the moves made by volunteers are valuable to the system.

In summary, the main contributions in this work are:

- We develop a practical scheme to deploy volunteers in a real time fashion to re-balance bikes in the system, using a combination of offline planning to deal with the uncertainties in the operating environment, and an online algorithm to guide the choices of the volunteers based on the real time status of the system.
- More importantly, we develop new insights on the performance of sparse structure in the bike re-balancing problem. We demonstrate that such structure can be used to concentrate re-balancing flows in a bike sharing system to reduce redundant moves (sometimes drastically), with only a small impact on the ability of the system to meet the demands of the riders.

Model

In this section, we formulate the problem of bike re-balancing using volunteers. We first describe a dynamic system of bike-sharing network with two streams of flows for riders and volunteers. We then embed the design of arc set for volunteer activities into decision making. Through model analysis, we show the impact of bike re-balancing structure, which leads us to focus on a tractable model. Then, we derive the moment model containing valuable sensitivity analysis information on arc selection and follow a recently proposed approach to construct the re-balancing structure.

We assume a bike-sharing network with N bikes, and each station (indexed by $\mathcal{I} = \{1, 2, \ldots, M\}$, where M is the total station number) is positioned with a certain number of bikes at the start of each day. For each station $i \in \mathcal{I}$, let $\tilde{A}_i(t)$ (resp. $\tilde{L}_i(t)$) denote the number of regular riders arriving to drop-off (resp. pick up) a bike at station i in time t. $\tilde{A}_i(t)$ and $\tilde{L}_i(t)$ are random but the distribution functions are known. The demand for regular riders here are the intrinsic demand in the market, and only some of these demands will be converted to rides, depending on the (empty) bike inventory distributions among different stations. i.e., there may be lost sales. For a typical station located in the central business district (CBD), in the early morning, more bikes will be returned to the station than picked up, and hence the system requires only a small number of target inventory (denoted as $TD_i(t)$ at station i, a.k.a., target demand level) initially. This target increases at the later part of the day, when the pick-up rate of bikes increases and dominates the return rate. In fact, the pioneering works in (Freund, Henderson, and Shmoys 2017; Freund et al. 2020) focus on finding the best target level of bikes at each station across time, through an innovative model measuring user dissatisfaction function.

We use the above information to develop the volunteer deployment scheme in the system. Consider the case when a volunteer arrives at time t. Let $N_i(t+T)$ denote the "*net number of bikes*" in station i at the time $t+T$, after accounting for the total number of bikes that flow in and out of the station (assuming all demands can be satisfied) during the interval $[t, t+T)$, and incorporating the final target inventory level $TD_i(t+T)$. Station i will be a pick-up station if $N_i(t+T) > 0$, and a drop-off station if $N_i(t+T) < 0$[1]. We define these terms formally later.

At the same time, we assume that there is a separate stream of volunteers arriving into the system, attracted by the incentive schemes offered by the system to move bikes from pick-up to drop-off stations. For ease of model analysis and without loss of generality, we assume that any re-balancing move by the volunteers at time t can be completed by time $t+T$. Note that in reality, the pool of the regular riders and volunteers may not be distinct, since some regular riders may decide to perform a re-balancing move (as a registered volunteer) if they are traveling from a pick-up station to a drop-off station anyway. We designate these rides as re-balancing moves in these cases. Also, unlike many other re-balancing studies, the re-balancing moves here are choices made by the volunteers and not dictated by the system. Hence some volunteers may arrive and leave without performing any re-balancing moves, if they could not find suitable moves. Besides, we do not consider the substitution

[1]For ease of exposition, we do not model neutral station in this paper.

effect directly, i.e., we assume regular riders will not look for bikes in nearby stations if they arrive at an empty station, and these demands will be lost sales to the system[2].

Instead of relying on station's status as pick-up or drop-off to decide whether to reward a move, suppose in addition we allow moves only on a (pre-selected) set of arcs in $\mathcal{E}$. This is in line with the observation by (He et al. 2020) that " treating bike stations as individual products is far from sufficient. The structure of the network – that is, where the connecting nodes are and what the weight is on each link—plays a significant role in determining the demand on each node." Assume that the volunteer arrival process to each station follows a Poisson process, and they choose to rebalance bikes for the system with a (thinned) Poisson process at a rate of $\gamma_{ij}(\mathcal{E})$ that may depend on the structure $\mathcal{E}$ [3]. Let $\tilde{t}_{ij}$ denote the amount of time s in the interval $(t, t+T]$ where a re-balancing move from station i to j will be rewarded by the system, based on the net number of bikes in the pair of stations. The number of (volunteer) moves along the arc (i,j) is therefore Poisson with mean

$$\tilde{e}_{ij}(\mathcal{E}) \sim Poi(\gamma_{ij}(\mathcal{E}) \times \tilde{t}_{ij}).$$

Let $\tilde{A}_i^{\mathcal{E}}(s)$ (resp. $\tilde{L}_i^{\mathcal{E}}(s)$) denote the induced number of volunteers arriving to drop-off (resp. pick up) a bike at station i in time s. We assume that the volunteer re-balancing activities will not lead to bike stock-out at the stations and change the usage statistics of the regular riders, since such re-balancing moves are controlled by the net number of bikes in the system. Incorporating the re-balancing moves by volunteers, the net number of bikes in each station i can be formally defined as follows:

$$\begin{aligned} N_i(t+T) = \quad & \overbrace{TD_i(t) - TD_i(t+T) + \int_t^{t+T} \left(\tilde{A}_i(s) - \tilde{L}_i(s)\right) ds}^{X_i(t+T)} \\ & + \int_t^{t+T} \left(\tilde{A}_i^{\mathcal{E}}(s) - \tilde{L}_i^{\mathcal{E}}(s)\right) ds. \end{aligned} \tag{1}$$

A re-balancing move from station i to station j in the interval $(t, t+T]$ is valuable if $X_i(t+T) > 0$ and $X_j(t+T) < 0$. Let

$$\tilde{c}_i := X_i^+(t+T),\ \tilde{d}_j := X_j^-(t+T)$$

denote the original imbalances in the system at time $t+T$ (without any re-balancing move by the volunteers). Let $\tilde{r} = (\tilde{c}, \tilde{d})$. Suppose the cost of a redundant move is λ and the saving from a valuable move is 1, then the aggregate value of restricting re-balancing moves in $\mathcal{E}$ is given by

$$\begin{aligned} Z_{\mathcal{E}}(\tilde{\mathbf{r}}, \tilde{\mathbf{e}}(\mathcal{E})) = \quad & \max\left(\sum_{(i,j)\in\mathcal{E}} \left\{ \min(\tilde{e}_{i,j}(\mathcal{E}), x_{ij}) - \lambda\left(\tilde{e}_{ij}(\mathcal{E}) - x_{ij}\right)^+ \right\} \right) \\ s.t. \quad & \sum_{i\in\mathcal{I},(i,j)\in\mathcal{E}} x_{ij} \le \tilde{d}_j \quad j \in \mathcal{I} \\ & \sum_{j\in\mathcal{I},(i,j)\in\mathcal{E}} x_{ij} \le \tilde{c}_i \quad i \in \mathcal{I} \\ & x_{ij} \ge 0 \quad (i,j) \in \mathcal{E} \end{aligned} \tag{2}$$

In this formulation, $x_{i,j}$ denotes the number of re-balancing moves from station i to j that are valuable for the system, when station i has surplus bikes at time $t+T$ (with $\tilde{c}_i > 0$) and station j has insufficient bikes (with $\tilde{d}_j > 0$). The constraints in (2) bound the total inflows for each demand node and the total outflows for each supply node under each specific realization. The above model could be used to capture the economical profitability of the bike-sharing system as follows: Suppose incentives for volunteer per move is λ, and value obtained from a regular ride is $1+\lambda$. So if a volunteer makes a valuable move, the value is $1 (= 1+\lambda-\lambda)$, and otherwise the move incurs a deployment cost of λ. By doing so, the objective of the current model setup could be directly used to measure the profitability of the system and guide the selection of arcs.

Using the fact that

$$\begin{aligned} & \min(\tilde{e}_{i,j}(\mathcal{E}), x_{ij}) - \lambda\left(\tilde{e}_{ij}(\mathcal{E}) - x_{ij}\right)^+ \\ = & (1+\lambda) \min\left\{ x_{ij}, \tilde{e}_{ij}(\mathcal{E}) \right\} - \lambda \tilde{e}_{ij}(\mathcal{E}) \end{aligned} \tag{3}$$

the above, with a slight abuse of notation, can be reformulated as:

$$\begin{aligned} Z_{\mathcal{E}}(\tilde{\mathbf{r}}, \tilde{\mathbf{e}}) = \quad & (1+\lambda) \max\left(\sum_{(i,j)\in\mathcal{E}} x_{ij} \right) - \lambda \sum_{(i,j)\in\mathcal{E}} \tilde{e}_{ij}(\mathcal{E}) \\ s.t. \quad & \sum_{i\in\mathcal{I},(i,j)\in\mathcal{E}} x_{ij} \le \tilde{d}_j \quad j \in \mathcal{I} \\ & \sum_{j\in\mathcal{I},(i,j)\in\mathcal{E}} x_{ij} \le \tilde{c}_i \quad i \in \mathcal{I} \\ & 0 \le x_{ij} \le \tilde{e}_{ij}(\mathcal{E}) \quad (i,j) \in \mathcal{E} \end{aligned} \tag{4}$$

Our problem reduces to finding the optimal solution $\mathcal{E}^*$ such that

$$\mathcal{E}^* := \text{argmax}_{\mathcal{E}} \mathbf{E}\left[Z_{\mathcal{E}}(\tilde{\mathbf{r}}, \tilde{\mathbf{e}}) \right]$$

Let

$$\begin{aligned} Z_{\mathcal{E}}^0(\tilde{\mathbf{r}}, \tilde{\mathbf{e}}) = \quad & \max\left(\sum_{(i,j)\in\mathcal{E}} x_{ij} \right) \\ s.t. \quad & \sum_{i\in\mathcal{I},(i,j)\in\mathcal{E}} x_{ij} \le \tilde{d}_j \qquad j \in \mathcal{I} \\ & \sum_{j\in\mathcal{I},(i,j)\in\mathcal{E}} x_{ij} \le \tilde{c}_i \qquad i \in \mathcal{I} \\ & 0 \le x_{ij} \le \tilde{e}_{ij}(\mathcal{E}) \qquad (i,j) \in \mathcal{E} \end{aligned} \tag{5}$$

Let $N(\mathcal{E})$ denote the total number of moves by volunteers in the interval $[t, t+T)$. Note that

$$\mathbf{E}\left[Z_{\mathcal{E}}(\tilde{\mathbf{r}}, \tilde{\mathbf{e}}) \right] = (1+\lambda) \mathbf{E}\left[Z_{\mathcal{E}}^0(\tilde{\mathbf{r}}, \tilde{\mathbf{e}}) \right] - \lambda \mathbf{E}\left[N(\mathcal{E}) \right]$$

The analysis of (4) is complicated because the random variable $\tilde{e}_{ij}(\mathcal{E})$ depends on $\mathcal{E}$ and the net number of bikes in each station. It is in general intractable.

At the same time, denote α and β as Lagrange multiplier vectors, by taking Lagrangian dual of the first two sets of constraints in (5), we have

$$\begin{aligned} Z_{\mathcal{E}}(\tilde{\mathbf{r}}, \tilde{\mathbf{e}}) = \quad & (1+\lambda) \min_{\alpha_i, \beta_j \ge 0} \quad \left\{ \sum_{i\in\mathcal{I}} \alpha_i \tilde{c}_i + \sum_{j\in\mathcal{I}} \beta_j \tilde{d}_j + Z(\alpha, \beta) \right\} \\ & -\lambda \sum_{(i,j)\in\mathcal{E}} \tilde{e}_{ij}(\mathcal{E}) \end{aligned} \tag{6}$$

[2] Trying to deploy volunteers to anticipate this substitution behavior will be quite challenging.

[3] Rate of volunteer arrival is independent of network structure. But the number of volunteers choosing to move bikes may be lower, due to network structure (i.e., choice restriction), and if there are outside options.

where

$$Z(\alpha,\beta) = \max\left(\sum_{(i,j)\in\mathcal{E}} \left\{ (1-\alpha_i-\beta_j)x_{ij} \right\} \right) \quad 0 \le x_{ij} \le \tilde{e}_{ij}(\mathcal{E}) \quad (i,j)\in\mathcal{E} \tag{7}$$

Note that in the optimal solution, $x_{i,j} = \tilde{e}_{ij}$ or 0 depending on the sign of $1-\alpha_i-\beta_j$. Hence

$$\begin{aligned} Z_{\mathcal{E}}(\tilde{\mathbf{r}},\tilde{\mathbf{e}}) = \; & (1+\lambda)\min\Big(\textstyle\sum_{i\in\mathcal{I}} \alpha_i \tilde{c}_i + \sum_{j\in\mathcal{I}} \beta_j \tilde{d}_j + \displaystyle\sum_{(i,j)\in\mathcal{E}} \Big\{ (1-\alpha_i-\beta_j)^+ \tilde{e}_{ij}(\mathcal{E}) \Big\} \Big) \\ & -\lambda \textstyle\sum_{(i,j)\in\mathcal{E}} \tilde{e}_{ij}(\mathcal{E}) \\ & \alpha_i, \beta_j \ge 0 \quad i,j\in\mathcal{I} \end{aligned} \tag{8}$$

Note that in the optimal solution, $\alpha_i, \beta_j \le 1$. We can therefore add this upperbound to the formulation (8) without affecting the optimal solution. We use this formulation to argue that a sparse structure may performs better than a fully flexible system.

Lower Bound. Using the fact that $(1-\alpha_i-\beta_j)^+ \ge 1-\alpha_i-\beta_j$, we have

$$\begin{aligned} & Z_{\mathcal{E}}(\tilde{\mathbf{r}},\tilde{\mathbf{e}}) \ge \\ & \textstyle\sum_{(i,j)\in\mathcal{E}} \tilde{e}_{ij}(\mathcal{E}) + (1+\lambda)\min_{\alpha_i,\beta_j\ge 0} \Big(\sum_{i\in\mathcal{I}} \alpha_i \Big(\tilde{c}_i - \sum_{j:(i,j)\in\mathcal{E}} \tilde{e}_{ij}(\mathcal{E}) \Big) \\ & \qquad + \textstyle\sum_{j\in\mathcal{I}} \beta_j \Big(\tilde{d}_j - \displaystyle\sum_{i:(i,j)\in\mathcal{E}} \tilde{e}_{ij}(\mathcal{E}) \Big) \Big) \\ & = \textstyle\sum_{(i,j)\in\mathcal{E}} \tilde{e}_{ij}(\mathcal{E}) - (1+\lambda) \Big(\sum_{i\in\mathcal{I}} \Big(\sum_{j:(i,j)\in\mathcal{E}} \tilde{e}_{ij}(\mathcal{E}) - \tilde{c}_i \Big)^+ \\ & \qquad + \textstyle\sum_{j\in\mathcal{I}} \Big(\sum_{i:(i,j)\in\mathcal{E}} \tilde{e}_{ij}(\mathcal{E}) - \tilde{d}_j \Big)^+ \Big). \end{aligned} \tag{9}$$

Now, in cases where $\tilde{e}_{ij}(\mathcal{E})$ is allowed only between station i and j with $\tilde{c}_i$ and $\tilde{d}_j$ are both large with high probability, we can ensure that the re-balancing moves will not be redundant and hence $Z_{\mathcal{E}}(\tilde{\mathbf{r}},\tilde{\mathbf{e}}) > 0$ almost surely. This ensures that choice restriction on some structure $\mathcal{E}$ will do as well as one without using volunteers. Thus crowd sourcing using volunteers will add value to the system in this environment.

Upper Bound. At the same time, let $Z^1_{\mathcal{E}}(\tilde{\mathbf{r}})$ denote $Z^0_{\mathcal{E}}(\tilde{\mathbf{r}},\tilde{\mathbf{e}})$ when $\tilde{\mathbf{e}}$ is replaced by ∞, i.e., by removing the upperbound on x_{ij} in (5). This is the classical max-flow problem on the structure $\mathcal{E}$. In this case, it is easy to see that

$$Z_{\mathcal{E}}(\tilde{\mathbf{r}},\tilde{\mathbf{e}}) \le (1+\lambda) Z^1_{\mathcal{E}}(\tilde{\mathbf{r}}) - \lambda N(\mathcal{E})$$

However, because

$$Z^1_{\mathcal{E}}(\tilde{\mathbf{r}}) \le \min\left(\sum_{i\in\mathcal{I}} \tilde{c}_i, \sum_{j\in\mathcal{I}} \tilde{d}_j \right),$$

increasing the density of the structure $\mathcal{E}$ by adding arcs may not help increase $Z^1_{\mathcal{E}}(\tilde{\mathbf{r}})$ as fast as increasing $N(\mathcal{E})$. In that case, using a dense structure may not perform as well as using a sparse structure!

In the rest of the paper, we focus on the problem by designing the right structure $\mathcal{E}$ to support the re-balancing operations in the crowd-sourced system. In particular, we analyze the impact of the structure $\mathcal{E}$ on the value $\mathbf{E}\left[Z^1_{\mathcal{E}}(\tilde{\mathbf{r}}) \right]$ instead, and use this upperbound to guide us to the design of a good structure for our problem.

Distributionally Robust Model

The main uncertainties stem from the time-varying net number of bikes in each station as well as the complicated interaction patterns among different stations. Hence we will use the first two moments information on $\tilde{c}$ and $\tilde{d}$ to characterize uncertainties and develop the moment model.

Recall that $\tilde{\mathbf{r}} = (\tilde{\mathbf{c}}, \tilde{\mathbf{d}})$, and

$$\begin{aligned} Z^1_{\mathcal{E}}(\tilde{\mathbf{r}}) = \; & \max_{x_{ij}} \Big(\sum_{(i,j)\in\mathcal{A}_0} x_{ij} \Big) \\ s.t. \; & \sum_{i\in\mathcal{I},(i,j)\in\mathcal{E}} x_{ij} \le \tilde{d}_j \qquad j\in\mathcal{I} \\ & \sum_{j\in\mathcal{J},(i,j)\in\mathcal{E}} x_{ij} \le \tilde{c}_i \qquad i\in\mathcal{I} \\ & x_{ij} \ge 0 \qquad (i,j)\in\mathcal{E} \end{aligned} \tag{10}$$

The support of $\tilde{\mathbf{r}}$ is non-negative, with $\tilde{c}_i = X_i^+$ and $\tilde{d}_i = X_i^-$ for some random variable X_i. This is a variant of the random maximum flow problem on the structure $\mathcal{E}$, with $\tilde{\mathbf{r}}$ not necessarily independent. By introducing dual variables $\mathbf{y}, \mathbf{z}$ for the two sets of constraints in (10), the dual of Problem (10) is

$$\begin{aligned} Z^1_{\mathcal{E}}(\tilde{\mathbf{r}}) = \; & \min_{\mathbf{y},\mathbf{z}} \sum_{j\in\mathcal{I}} \tilde{d}_j y_j + \sum_{i\in\mathcal{I}} \tilde{c}_i z_i \\ s.t. \; & y_j + z_i \ge 1 \quad (i,j)\in\mathcal{E} \\ & \mathbf{z} \ge \mathbf{0} \\ & \mathbf{y} \ge \mathbf{0} \end{aligned} \tag{11}$$

When $\tilde{\mathbf{c}}$ represents fixed capacity and $\tilde{\mathbf{d}}$ represents random demand in a supply chain system, the above problem has been thoroughly analyzed in (Yan, Gao, and Teo 2018) using a distributionally robust reformulation of the problem with only the first two moments information. Moreover, when the system is balanced, i.e. $\tilde{c}_i = \mu = \mathbf{E}[\tilde{d}_j]$, they show that the distributionally robust model under a 2-chain has almost the same expected value as the fully flexible system, confirming a well-known observation in the Process Flexibility literature and the references therein for details). We extend their technique to analyze (11), and show that similar observation holds even in a transshipment model when $\tilde{\mathbf{c}}$ and $\tilde{\mathbf{d}}$ may not be independent. Meanwhile, we would like to highlight the key differences of the model used in our bike rebalancing setting as follows: The model by (Yan, Gao, and Teo 2018) solves a bipartite matching problem, with uncertainty on only one side. We borrow this technique to solve a variant of the transshipment problem, with uncertainty on both sides. In doing so, several features of the transshipment model can be incorporated to the basic approach to improve the performance and obtain a better structure. We also embed the feature that no node can be supply and demand node at the same time in our moment model (16).

We discuss next how Model (11) can be reformulated as a quadratic constrained problem based on the totally unimodular (TU) property of the constraint matrix. Note that the optimal dual solution is 0-1 in our model, so we can replace $y_j + z_i \ge 1$ by the following quadratic terms:

$$(1-y_j)(1-z_i) \;=\; 0 \quad (i,j)\in\mathcal{E} \tag{12}$$

We obtain an equivalent reformulation of (11) as follows:

$$
\begin{array}{llcl}
 & Z_1(\tilde{\mathbf{r}}) = \min\limits_{\mathbf{y},\mathbf{z},\alpha} \sum\limits_{j\in\mathcal{J}} \tilde{d}_j y_j + \sum\limits_{i\in\mathcal{I}} \tilde{c}_i z_i & & \\
s.t. & (1-y_j)(1-z_i) & = 0 & (i,j)\in\mathcal{E} \\
 & \mathbf{y}\in\{0,1\}^n, \mathbf{z}\in\{0,1\}^n & &
\end{array}
\tag{13}
$$

Following (Yan, Gao, and Teo 2018), this can be written in a more general quadratic program, with $\tilde{\mathbf{r}} = (\tilde{\mathbf{c}}, \tilde{\mathbf{d}})$ and $\mathbf{x} = \{x_{ij}, \forall i \in \mathcal{I}, j \in \mathcal{I}\}$:

$$
\begin{array}{lrcl}
 & Z(\tilde{\mathbf{r}}) = \min\limits_{\mathbf{x}} & & \tilde{\mathbf{r}}^\top \mathbf{x} \\
s.t. & \mathbf{a}_i^\top \mathbf{x} & = & b_i, \forall i \\
 & (\mathbf{h}_i^\top \mathbf{x} + f_i)(\hat{\mathbf{h}}_j^\top \mathbf{x} + \hat{f}_j) & = & 0, \forall (i,j) \in \mathcal{H} \\
 & x_i & \in & \{0,1\}, \forall i \in \mathcal{B}
\end{array}
\tag{14}
$$

where $\mathcal{B}$ is the set of indices for the binary variables. $\mathcal{H}$ is the subset associated with linking arcs in the network, i.e., $\mathcal{H} \subseteq \mathcal{E}$. We assume the dimension of decision vector is $\mathcal{N}$, i.e. $\mathbf{x} \in \mathbf{R}^{\mathcal{N}}$, and the number of linear constraints in (14) is $\mathcal{M}$.

Assuming only the first-two moments information of $\tilde{\mathbf{r}}$ are available, i.e., finite first moment μ_r and finite second moment Σ_r, the problem of interest is shown as follows:

$$
(P) \quad Z_P = \inf_{\tilde{\mathbf{r}} \sim (\mu_{\mathbf{r}}, \boldsymbol{\Sigma}_{\mathbf{r}})} \mathbf{E}\big[Z[\tilde{\mathbf{r}}]\big] \tag{15}
$$

Using the lifted variables $\mathbf{p}$, X, and Y as the decision variables, we obtained the following completely positive program to (15):

$$
\begin{array}{lrcll}
(C) \quad Z_C = \min & I \bullet Y & & & \\
s.t. & \mathbf{a}_i^\top \mathbf{p} & = & b_i, & \forall i = 1, ..., \mathcal{M} \\
 & \mathbf{a}_i^\top X \mathbf{a}_i & = & b_i^2, & \forall i = 1, ..., \mathcal{M} \\
 & X_{ii} & = & p_i, & \forall i \in \mathcal{B} \\
 & \mathbf{h}_i^\top X \hat{\mathbf{h}}_j + (f_i \hat{\mathbf{h}}_j^\top + \hat{f}_j \mathbf{h}_i^\top)\mathbf{p} + f_i \hat{f}_j & = & 0, & \forall (i,j) \in \mathcal{H} \\
 & \begin{pmatrix} 1 & \mu_{\mathbf{r}}^\top & \mathbf{p}^\top \\ \mu_{\mathbf{r}} & \Sigma_r & Y^\top \\ \mathbf{p} & Y & X \end{pmatrix} & \succeq_{cp} & 0 &
\end{array}
\tag{16}
$$

From the construction of this completely positive program (16), which is named the moment model in this paper, it is clear that $Z_C \geq Z_P$ since it is a relaxation of problem (15).

Proposition 1 *Under appropriate technical conditions, the completely positive program Z_C and the worst-case model Z_P are equivalent, i.e. $Z_C = Z_P$.*

We used the above conic program to design sparse structure for the transshipment problem. The main idea can be briefly demonstrated as follows: Starting from a full graph $\mathcal{E}^F$ with all possible arcs, the heuristic incrementally deletes the arc with the smallest absolute value in the optimal dual solution. At iteration k, the network is denoted as $\mathcal{E}^k$. The dual-variable-based heuristic is detailed in (Yan, Gao, and Teo 2018) as well as the technical appendix. As our model fits into this general framework, we use the same deletion heuristic to construct our network.

Deploying Volunteers via Sparse Network

Note that the real time re-positioning of empty bikes is a notoriously difficult problem. Earlier work has focused on state-independent control policy using fluid approximation models (Banerjee, Freund, and Lykouris 2016; Braverman et al. 2019). However, this method requires stringent assumptions on the dynamics of the stochastic systems and produces control policies that are static and could not fully capitalize on the dynamic real time information available in the system. Integrating the two streams of arrival processes (one for riders, another for volunteers) appear to be technically challenging using these methods. We use here instead a state-dependent control policy to re-position bikes to make them available to riders, and use the sparse network to concentrate the flows on "relevant" OD pairs.

The deployment policy we proposed is motivated by the practice in Citi Bike, through the identification of pick-up and drop-off stations. However, we allow only moves along arcs in the sparse structure identified, to concentrate the re-balancing moves on essential arcs only. We use a target inventory level at each station, and use the (random) arrival and departure at each station, over the designated time interval, to construct our structure. We envisage that a structure will be used for each of the time intervals - $[0, T], [T, 2T], [2T, 3T] \ldots$ til the end of the day. For simplicity, we assume that the demand is stationary across periods, and hence the structure remains unchanged across periods. For dynamic problem, the structure of each time segment can vary based on the change in the arrival and departure processes.

Basic Setting

We consider a bike sharing system with M stations and N bikes. The set of stations is denoted by $S = \{s_1, s_2, \cdots, s_M\}$. We distinguish the bike flow by regular riders (i.e., users) and volunteers. At $t = 0$, all stations started with certain number of (empty) bikes. The riders arrive to each station i following a Poisson process with exogenous rate λ_i. For a rider arriving at station i, she will pick up an empty bike and ride to station j with probability P_{ij}. If no empty bike is available, the rider will leave the system and the system experiences a *lost sale*. Note that we allow $P_{ii} > 0$ to incorporate the trips starting and ending at the same station i. We assume that travel times from station i to j are i.i.d. exponential random variables with mean $1/\mu_{ij}$. When the rider finishes her trip at station j, the bike is parked and becomes available to the next rider. The goal of the bike sharing platform is to re-position the empty bikes properly across all the stations to minimize lost sales (i.e., maximize number of usage or demand fulfilled).

Formally, let τ denote the simulation horizon (i.e., simulation for day $d \in \mathbb{D} = \{1, 2, \cdots, \tau\}$), and the data in each day is aggregated over κ disjoint periods (i.e., period $t' \in \mathbb{T} = \{1, 2, \cdots, \kappa\}$). It is assumed all time periods are the aggregation of time t within period t'. The number of riders arriving at station i in period t' on day d is denoted by $D_i^d(t')$.

- The "Target Demand Level" in station i at period t' is denoted by

$$
TD_i(t') := \bar{D}_i(t'), \text{ where } \bar{D}_i(t') = \sum_{d=1}^{\tau} D_i^d(t')/\tau. \tag{17}
$$

We suppress the dependence of TD_i on the period t' if demand is homogeneous across time, but we allow them to be heterogeneous across stations.

- We monitor the number of bikes in each station in real time (denoted by $\tilde{X}_i(t)$), and the target demand level $TD_i(t)$ at time t is simply $TD_i(t')$ if t is in the time period t'. In this way, we define the real-time "net number of bikes" $\tilde{N}_i(t)$ as follows:

$$\tilde{N}_i(t) = \tilde{X}_i(t) - TD_i(t). \tag{18}$$

where $TD_i(t) = TD_i(t'), \forall t \ in \ t', t' \in \mathbb{T} = \{1, 2, \dots, \kappa\}$.

Remark. When a volunteer arrives at time t into the system, station i is a pick-up station if $\tilde{N}_i(t) > 0$, a drop-off station if $\tilde{N}_i(t) < 0$, a neutral station if $\tilde{N}_i(t) = 0$.

Volunteer Deployment

We assume that the volunteers arrive into each station i following a Poisson process with rate $\lambda_i^{(v)}$. For simplicity, we assume $\lambda_1^{(v)} = \lambda_2^{(v)} = \cdots = \lambda_M^{(v)} = \lambda^{(v)}$ in our simulation. Denote volunteer arrival time in the ascending order as $v_1 \leq v_2 \leq v_3 \leq \cdots \leq v_L$, where L is the index of last volunteer. Let $\sigma(v_l)$ denote the station in which the lth volunteer shows up at time v_l. Each time a volunteer arrives, if the station has surplus bikes (i.e., a pick-up station), the platform will next detect a set of drop-off stations based on the current distribution of empty bikes (excluding those in transit). Our online volunteer deployment scheme, elaborated in Algorithm 1, will further restrict the drop-off options based on the network structure $\mathcal{E}$ constructed earlier, and reward the volunteers only for moves sanctioned by the algorithm. We assume the trip duration for the volunteers are identical to that of the riders. Let $S_t^{(r)}$ denote the set of drop-off stations available for bike repositioning at time t and $\mathcal{O}$ denote the attractiveness of the outside option[4]. The volunteer will move an empty bike to a drop-off station with probability $\frac{1}{|S_t^{(r)}|+\mathcal{O}}$ when $S_t^{(r)} \neq \emptyset$. The number of "effective" volunteers to each station will depend on the choice sets of available drop-off stations dynamically offered by the algorithm.

Performance Measure

At time t, let $E_{ij}(t)$ denote the number of empty bikes en route from station i to station j ($j \neq i$) (i.e., re-balancing moves by volunteers), and let $E_{ii}(t)$ be the number of empty bikes located at station i. Let $E(t)$ be the $M \times M$ matrices whose $(i,j)th$ entries are $E_{ij}(t)$. We use the total fulfilled demand $\mathbb{P}_1$ and the total re-balancing moves $\mathbb{P}_2$ as the performance measures for this study. Formally, let $V' = \{v'_1, v'_2, v'_3, \cdots, v'_{L'}\}$ denote the set of arrival times for the riders, with $v'_1 \leq v'_2 \leq v'_3 \leq \cdots \leq v'_{L'}$, where L' denotes the index for last rider. Let $\sigma(v'_l)$ denote the station in which the lth rider shows up at time v'_l, so we have $\sigma(v'_l) \in S$. We define:

$$\mathbb{P}_1 \triangleq \sum_{l=1}^{L'} \mathbb{I}\{E_{\sigma(v'_l)\sigma(v'_l)}(v'_l) > 0\} \tag{19}$$

Here (19) reflects the availability of empty bikes for demand fulfillment when a rider arrives. Similarly, by monitoring the arrival process of the volunteers, we have

$$\mathbb{P}_2 \triangleq \sum_{l=1}^{L} \mathbb{I}\{S_{v_l}^{(r)} \neq \emptyset, \tilde{N}_{\sigma(v_l)}(v_l) \geq 0\} \tag{20}$$

Here (20) counts the number of re-balancing moves made by the volunteers.

[4]Note that the random behavior of the volunteers could be modeled more accurately by using Multinomial Logit (MNL) choice framework, based on the rewards and other important features.

Algorithm 1: State-dependent Empty Bike Re-balancing Policy

Step 1. Set the bike re-positioning network to be a predetermined graph, i.e. $\mathcal{E} = \mathcal{E}^F$ or $\mathcal{E} = \mathcal{E}^{R^*}$, where $\mathcal{E}^{R^*}$ denotes the optimal sparse structure obtained by our method. Set $l = 1$.

Step 2. For the lth volunteer showing up at station $\sigma(v_l) \in S$, check the station status $\tilde{N}_{\sigma(v_l)}$ (i.e., the net number of bikes accounting for the target number of bikes for this station). Stop if no volunteer comes.

Step 3. If $\tilde{N}_{\sigma(v_l)}(v_l) \leq 0$, station $\sigma(v_l)$ is a drop-off station or neutral station, set $S_{v_l}^{(r)} = \emptyset, l = l+1$, go to Step 2. If $\tilde{N}_{\sigma(v_l)}(v_l) > 0$, station $\sigma(v_l)$ is a pick-up station, set $S_{v_l}^{(r)} = \{i | \tilde{N}_i(v_l) < 0, i \in S \setminus \{\sigma(v_l)\}, (\sigma(v_l), i) \in \mathcal{E}\}$, and go to Step 4.

Step 4. If $S_{v_l}^{(r)} = \emptyset$, set $l = l+1$, go to Step 2. If $S_{v_l}^{(r)} \neq \emptyset$, we assume the lth volunteer will randomly pick one station $k \in S_{v_l}^{(r)}$ to re-position the empty bike with probability $\frac{1}{|S_{v_l}^{(r)}|+\mathcal{O}}$, set $l = l+1$, then go to Step 2.

Case Study: Boston Hubway System

In the rest of the paper, we evaluate the performance of our technique for the bike sharing system in Boston. The data for the Boston Hubway system spans around three months from May 1, 2012 to July 9, 2012, with approximately 60 base stations in operations. First, we use the 50-weekday usage data from the Boston Hubway System to estimate the parameters in our model, based on the morning commutes (i.e., 7:00-13:00) (c.f., technical appendix). Then we used these parameters to generate simulation data including rider trip commuting pattern, over a 6-hour time window (i.e., 36 ten-minute time slots/36 periods). We use 10 minutes to denote one period. We also assume $N = 300$ bikes and $N = 600$ bikes in our simulation. We first run the simulation when no volunteer is available over 50 days and 100 days respectively, after which we use the 50-day output as training data (i.e., in-sample) to design the sparse network, and test our online re-balancing algorithm over another new 100-day simulation.

Hubway Network Design: Offline Problem

In the technical appendix, we describe how we prepare the input to our distributionally robust model, based on the train-

$\mathcal{E}^F(3540)$	$\mathcal{E}^R(720)$	$\mathcal{E}^R(480)$	$\mathcal{E}^R(240)$	$\mathcal{E}^R(120)$
113.9737	113.4153	112.9123	111.8150	109.4021

Table 1: Worst-case Expected Maximum Flow under Different Structures Using Network Design Model

In-Sample Performance		Out-Sample Performance	
$\mathcal{E}$	Gap	$\mathcal{E}$	Gap
130.46 (120-arc)	2.71 %	126.36 (120-arc)	4.45%
132.38 (240-arc)	1.28%	129.50 (240-arc)	2.08%
133.10 (480-arc)	**0.75%**	**130.53 (480-arc)**	**1.30%**
133.54 (720-arc)	0.42%	131.15 (720-arc)	0.83%
134.10 (3540-arc)	-	132.25 (3540-arc)	-

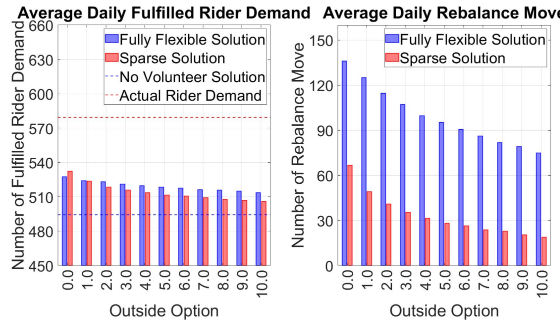
Table 2: Simulation Performance for Hubway BSS

ing data available. In the rest of this section, we apply the dual-variable-based heuristic to construct different network structure. Note that for larger problems, we can divide the city into zones. We solve semidefinite programming with moderate size in each iteration of this heuristic using the state-of-the-art SDPNAL+ solver developed by (Yang, Sun, and Toh 2015). We evaluate the performance of the network using simulation results based on the training and testing data set, to demonstrate that the sparse network structure obtained using our methodology performs well in practice.

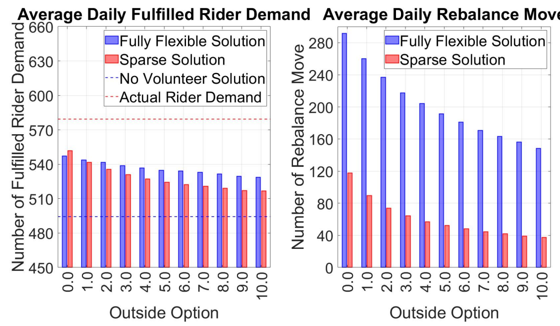
We obtained four specific sparse structures (i.e., 120-arc, 240-arc, 480-arc, and 720-arc) to demonstrate how dramatic change in performance might occur. Table 1 compares the worst-case expected maximum flow value under the fully flexible structure $\mathcal{E}^F(3540)$ and these sparse structures. The performance of the sparse structure with 480 arcs is already very close to the performance under the fully flexible structure (gap of only 0.93%). To further evaluate the performance of the sparse structures, we compare in Table 2 the maximum flow for fully and sparse structure by solving the linear program (11) for each structure. Table 2 shows the performance of the in-sample and out-sample expected maximum flow, in which the gap is defined as $\frac{|\mathcal{E}^F - \mathcal{E}^R|}{\mathcal{E}^F} * 100\%$. The results suggest that the sparse structure with 480 arcs can perform almost as well as the fully flexible structure.

Online Deployment in Hubway

The previous section evaluates the performance of the structure in the offline model. In what follows, we demonstrate that these structures make a difference in the performance of the online deployment algorithm, using the 100-day testing set in the online setting. Without volunteer participation when $N = 600$, we observe that the usage of volunteers will not increase the demand fulfilled by too much in this case, since there is already very few lost sales. However, when we have only $N = 300$ bikes in the system, it turns out that the volunteer-based bike re-balancing scheme has a



(a) Volunteer Poisson Arrival Parameter $c = 0.05$



(b) Volunteer Poisson Arrival Parameter $c = 0.1$

Figure 1: Performance Comparison between Sparsity Solution and Fully Solution in Hubway BSS (480arc)

huge impact. Note that there were around 580 requests for bikes each day, and without volunteers, the 300 bikes can support around 495 rides while the 600 bikes can support around 570 rides. In the following, we will focus on the 300-bike system and reveal how volunteer participation improves the rider demand fulfillment.

Using a similar setup, we model the arrival of volunteers to the system using a separate Poisson process. We set the homogeneous volunteer arrival rate for each station to be $c \times \max\{\lambda_i^{(v)}, \forall i \in S\}$, where c is 0.05 (around 175 volunteers) or 0.1 (around 349 volunteers). The outside option $\mathcal{O}$ varies from 0 to 10 with step size 1. The performance is as shown in Fig 1. As observed, when no outside option exists, the on-line deployment algorithm using the sparse system is able to reduce re-balancing moves with a drastic drop against fully flexible system for essentially nearly the same level of demand fulfilled. The same conclusion holds even when the outside option $\mathcal{O}$ increases from 0 to 10.

Concluding Remark

Motivated by the Bike Angels program in New York's Citi Bikes, we study the crowd-sourced bike re-balancing problem. We solved this problem using a recent method proposed by (Yan, Gao, and Teo 2018) to construct the backbone for the bike re-balancing problem. We use the sparse network to guide the re-balancing move of the volunteers. Our numerical studies show that this can be a very effective and low cost solution for the re-balancing operation in these systems.

Acknowledgments

This research is supported by the Ministry of Education, Singapore, under its 2019 Academic Research Fund Tier 3 grant call (Award ref: MOE-2019-T3-1-010)

References

Aeschbach, P.; Zhang, X.; Georghiou, A.; and Lygeros, J. 2015. Balancing Bike Sharing Systems Through Customer Cooperation-A Case Study on London's Barclays Cycle Hire. In *2015 54th IEEE Conference on Decision and Control (CDC)*, 4722–4727. IEEE.

Alvarez-Valdes, R.; Belenguer, J. M.; Benavent, E.; Bermudez, J. D.; Muñoz, F.; Vercher, E.; and Verdejo, F. 2016. Optimizing the Level of Service Quality of a Bike-Sharing System. *Omega* 62: 163–175.

Angeloudis, P.; Hu, J.; and Bell, M. G. H. 2014. A Strategic Repositioning Algorithm for Bicycle-Sharing Schemes. *Transportmetrica A: Transport Science* 10(8): 759–774.

Banerjee, S.; Freund, D.; and Lykouris, T. 2016. Pricing and Optimization in Shared Vehicle Systems: An Approximation Framework. *arXiv preprint arXiv:1608.06819* .

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.

Bonnotte, N.; Cherrier, R.; Delassus, R.; and Alouini, Y. 2015. Real-Time Data Analytics and Optimization of Shared-Vehicles Networks. In *2nd ITS World Congres*.

Braverman, A.; Dai, J. G.; Liu, X.; and Ying, L. 2019. Empty-car Routing in Ridesharing Systems. *Operations Research* 67(5): 1437–1452.

Cagliero, L.; Cerquitelli, T.; Chiusano, S.; Garza, P.; and Xiao, X. 2017. Predicting Critical Conditions in Bicycle Sharing Systems. *Computing* 99(1): 39–57.

Chemla, D.; Meunier, F.; Pradeau, T.; Calvo, R. W.; and Yahiaoui, H. 2013. Self-Service Bike Sharing Systems: Simulation, Repositioning, Pricing .

Chou, M. C.; Chua, G. A.; Teo, C. P.; and Zheng, H. 2011. Process Flexibility Revisited: The Graph Expander and its Applications. *Operations Research* 59(5): 1090–1105.

Chung, H.; Freund, D.; and Shmoys, D. B. 2018. Bike Angels: An Analysis of Citi Bike's Incentive Program. In *Proceedings of the 1st ACM SIGCAS Conference on Computing and Sustainable Societies*, 1–9.

Freund, D.; Henderson, S. G.; and Shmoys, D. B. 2017. Minimizing multimodular functions and allocating capacity in bike-sharing systems. In *International Conference on Integer Programming and Combinatorial Optimization*.

Freund, D.; Norouzi-Fard, A.; Paul, A.; Wang, C.; Henderson, S. G.; and Shmoys, D. B. 2020. Data-driven Rebalancing Methods for Bike-share Systems. In *Analytics for the Sharing Economy: Mathematics, Engineering and Business Perspectives*, 255–278. Springer.

Fricker, C.; and Gast, N. 2016. Incentives and Redistribution in Homogeneous Bike-Sharing Systems with Stations of Finite Capacity. *Euro Journal on Transportation and Logistics* 5(3): 261–291.

Froehlich, J.; Neumann, J.; and Oliver, N. 2008. Measuring the Pulse of the City through Shared Bicycle Programs. *Proc. of UrbanSense08* 16–20.

Ghosh, S.; Jing, Y. K.; and Patrick, J. 2019. Improving Customer Satisfaction in Bike Sharing Systems through Dynamic Repositioning. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19)*.

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)*.

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.

He, P.; Zheng, F.; Belavina, E.; and Girotra, K. 2020. Customer Preference and Station Network in the London Bike-Share System. *Management Science* .

Jian, G.; Hugo, N.; and Lu, L. 2019. Robust Bike-sharing Stations Allocation and Path Network Design: A Two-stage Stochastic Programming Model. *Transportation Letters* 1–10.

Kaltenbrunner, A.; Meza, R.; Grivolla, J.; Codina, J.; and Banchs, R. 2010. Urban Cycles and Mobility Patterns: Exploring and Predicting Trends in a Bicycle-based Public Transport System. *Pervasive and Mobile Computing* 6(4): 455–466.

Kloimüllner, C.; and Raidl, G. R. 2017. Full-load Route Planning for Balancing Bike Sharing Systems by Logic-based Benders Decomposition. *Networks* 69(3): 270–289.

Lefkowitz, M. 2018. With Real-time Decisions, Citi Bike Breaks the Cycle of Empty Stations. *Cornell Chronicle* .

Li, Y.; Szeto, W.; Long, J.; and Shui, C. S. 2016. A Multiple Type Bike Repositioning Problem. *Transportation Research Part B: Methodological* 90: 263–278.

Liu, J.; Sun, L.; Chen, W.; and Xiong, H. 2016. Rebalancing Bike Sharing Systems: A Multi-Source Data Smart Optimization. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1005–1014.

Lu, C. 2016. Robust Multi-period Fleet Allocation Models for Bike-sharing Systems. *Networks and Spatial Economics* 6(1): 61–82.

McGowen, M. 2018. Reimagining Bike Sharing: Cornell Researchers Develop Smoother Bike Transport. *The Cornell Daily Sum* .

O'Mahony, E.; and Shmoys, D. B. 2015. Data Analysis and Optimization for (Citi) Bike Sharing. In *Twenty-ninth AAAI Conference on Artificial Intelligence*. Citeseer.

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.

Schuijbroek, J.; Hampshire, R. C.; and Van Hoeve, W. J. 2017. Inventory Rebalancing and Vehicle Routing in Bike Sharing Systems. *European Journal of Operational Research* 257(3): 992–1004.

Shahsavaripour, S. 2015. Different Methods of Redistributing Bike among the Station: A Mathematical Solution in Order to Find Critical Stations. *American Journal of Environmental Engineering* 5(1): 21–31.

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*, volume 1, 723–729.

Yan, Z.; Gao, S. Y.; and Teo, C. P. 2018. On the Design of Sparse but Efficient Structures in Operations. *Management Science* 64(7): 3421–3445.

Yang, L.; Sun, D.; and Toh, K. C. 2015. SDPNAL+: A Majorized Semismooth Newton-CG Augmented Lagrangian Method for Semidefinite Programming with Nonnegative Constraints. *Mathemtical Programming Computation* 7: 1737–1765.

Zhang, D.; Yu, C.; Desai, J.; Lau, H.; and Srivathsan, S. 2017. A Time-Space Network Flow Approach to Dynamic Repositioning in Bicycle Sharing Systems. *Transportation Research Part B: Methodological* 103: 188–207.