

# Dynamic Redeployment to Counter Congestion or Starvation in Vehicle Sharing Systems

Supriyo Ghosh, Pradeep Varakantham, Yossiri Adulyasak<sup>†</sup>, Patrick Jaillet<sup>‡</sup>

School of Information Systems, Singapore Management University, Singapore 178902

<sup>†</sup>Singapore-MIT Alliance for Research and Technology (SMART), Massachusetts Institute of Technology

<sup>‡</sup>Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology  
supriyog.2013@phdis.smu.edu.sg, pradeepv@smu.edu.sg, yossiri@smart.mit.edu, jaillet@mit.edu

## Abstract

Extensive usage of private vehicles has led to increased traffic congestion, carbon emissions, and usage of non-renewable resources. These concerns have led to the wide adoption of vehicle sharing (ex: bike sharing, car sharing) systems in many cities of the world. In vehicle-sharing systems, base stations (ex: docking stations for bikes) are strategically placed throughout a city and each of the base stations contain a pre-determined number of vehicles at the beginning of each day. Due to the stochastic and individualistic movement of customers, there is typically either congestion (more than required) or starvation (fewer than required) of vehicles at certain base stations. As demonstrated in our experimental results, this happens often and can cause a significant loss in demand. We propose to dynamically redeploy idle vehicles using carriers so as to minimize lost demand or alternatively maximize revenue for the vehicle sharing company. To that end, we contribute an optimization formulation to jointly address the redeployment (of vehicles) and routing (of carriers) problems and provide two approaches that rely on decomposability and abstraction of problem domains to reduce the computation time significantly. Finally, we demonstrate the utility of our approaches on two real world data sets of bike-sharing companies.

## 1 Introduction

Shared Transportation Systems (STS) offer attractive alternatives to deal with serious concerns of private transportation such as increased carbon emissions, traffic congestion and usage of non-renewable resources. Popular examples of STS are bike sharing (ex: Capital Bikeshare in Washington DC, Hubway in Boston, Bixi in Montreal, Velib in Paris, Wuhan and Hangzhou Public Bicycle in Hangzhou) and car sharing (ex: Car2go in Seattle, Zipcar in USA) systems, which are installed in many major cities around the world. Bike sharing systems are widely adopted with 747 active systems, a fleet of over 772,000 bicycles and 235 systems in planning or under construction (Meddin and DeMaio 2015). A bike-sharing system (BSS) typically has a few hundred base stations scattered throughout a city. At the beginning of the day, each station is stocked with a pre-determined

number of bikes. Users with a membership card can pickup and return bikes from any designated station, each of which has a finite number of docks. At the end of the work day, carrier vehicles (ex: trucks) are used to move bikes around so as to return to some pre-determined configuration at the beginning of the day.

Due to the individual movement of customers according to their needs, there is often congestion (more than required) or starvation (fewer than required) of bikes on aggregate at certain base stations. Figure 1 provides the number of instances when stations become empty or full during the day (summed over each month) for a leading bike-sharing company which performs redeployment of bikes to stations only once at the end of the day. A full station can be considered as being indicative of congestion and an empty station can be considered as being indicative of starvation. At a minimum, there are around 100 cases of empty stations and 100 cases of full stations per day and at a maximum there are about 750 cases of empty stations and 330 cases of full stations per day. This serves as the motivation for this paper, where we employ dynamic redeployment during the day to better match demand with supply.

As demonstrated in (Fricker and Gast 2012) and our experimental results, this (particularly starvation) can result in a significant loss of customer demand. Such loss in demand can have two undesirable outcomes: (a) loss in revenue; (b) increase in carbon emissions, as people can resort to fuel burning modes of transport. So, there is a practical need to minimize the lost demand and our approach is to dynamically redeploy bikes with the help of carriers (typically medium to large sized trucks) during the day. However, because carriers incur a cost in performing redeployment, we have to consider the trade-off between minimizing lost demand (alternatively maximizing revenue) and cost of using carriers for redeployment. Henceforth, we refer to this problem as the Dynamic Redeployment and Routing Problem (DRRP).

Minor variations of DRRP are applicable to more general shared transportation systems, empty vehicle redistribution in Personal Rapid Transit (PRT) (Lees-Miller, Hammersley, and Wilson 2010) and dynamic redeployment of emergency vehicles (Yue, Marla, and Krishnan 2012; Saisubramanian, Varakantham, and Chuin 2015).

Given the practical benefits of bike sharing systems and

the challenging nature of setting up such systems to operate efficiently, there have been a wide variety of research papers addressing the problem of lost demand and other issues pertinent to it. The key distinction from existing research on bike sharing is that we consider the dynamic redeployment of bikes in conjunction with the routing problem for carriers.

DRRP is an NP-Hard problem and therefore, we focus on principled approximations. Specifically, our key contributions are as follows :

(1) A mixed integer and linear optimization formulation to maximize profit for the bike sharing company by trading off between:

- computing the optimal re-deployment strategy (i.e., how many vehicles have to be picked or dropped from each base station and when) for bikes; **and**
- computing the optimal routing policy (i.e., what is the order of base stations according to which redeployment happens) for each of the carriers.

(2) A Lagrangian dual decomposition method to exploit the weak dependency between the component which computes re-deployment strategy for bikes and the component which computes routing policy for carriers.

(3) An abstraction mechanism that groups nearby base stations to reduce the size of the decision problem and consequently, improve scalability.

Extensive computational results on real-world datasets of two bike-sharing companies, namely Capital Bikeshare (Washington, DC) and Hubway (Boston, MA) demonstrate that our techniques improve revenue and operational efficiency of bike-sharing systems.

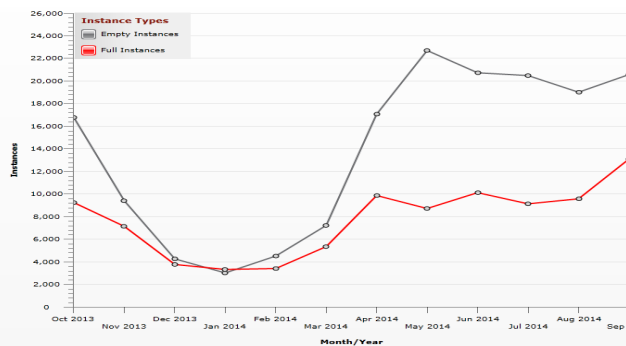


Figure 1: Number of instances of empty/full stations in CapitalBikeShare Company

## 2 Related Work

Given the practical benefits of bike sharing systems, they have been studied extensively in the literature. We focus on three threads of research that are of relevance to this paper. First thread of papers focus on the problem of finding routes at the end of the day for a fixed set of carriers to achieve the desired configuration of bikes across the base stations. (Schuijbroek, Hampshire, and van Hoes 2013; Raviv and Kolka 2013; Raviv, Tzur, and Forma 2013; Rainer-Harbach et al. 2013) have provided scalable exact and approximate approaches to this routing problem by either abstracting base stations into mega stations or by employing insights from

inventory management or by using variable neighborhood search based heuristics. All the papers in this thread assume there is only one fixed redeployment of bikes that happens at the end of the day. In contrast, our approaches focus on dynamic redeployment(s) during the day.

The second thread of research focuses on the placement of base stations and on performing dynamic redeployment of bikes during the day. (Shu et al. 2013; 2010) predict the stochastic demand from user trip data of Singapore metro system using poisson distribution and provide an optimization model that suggests the best location of the stations and a dynamic redeployment model to minimize the number of unsatisfied customers. However, they assume that redeployment of bikes from one station to another is always possible without considering the routing of carriers, which is a major cost driver for the bike-sharing company. A dynamic redeployment model was proposed in (Contardo, Morency, and Rousseau 2012) to deal with unmet demand in rush hours. They provide a myopic redeployment policy by considering the current demand. They employed Dantzig-Wolfe and Benders decomposition techniques to make the decision problem faster. (Pfrommer et al. 2014) also provides a myopic online decisions based on assessment of demand for the next 30 minutes. As can be observed from the data, customer demand of bikes varies over time stochastically and hence a myopic redeployment policy can significantly falter and may lead to circular movements for the carriers as it does not consider the future demand. Our approaches differ from this thread of research as we consider dynamic redeployment and routing of carriers together and consider the multi-step expected demand in determining the dynamic redeployment policy.

The third thread of research which is complementary to the work presented in this paper is on demand prediction and analysis. (Nair and Miller-Hooks 2011) provides a service level analysis of the Bike Sharing System using a dual-bounded joint-chance constraints where they predict the near future demands for a short period of time. While, this may not be applicable for a large system with a small set of carriers, the insights generated are practical and useful in demand prediction. (Leurent 2012) represent the bike sharing system as a dual markovian waiting system to predict the actual demand. As we already highlighted, given its generality and applicability over an entire horizon, we also employ the demand prediction model by (Shu et al. 2013; 2010) and assume that demand follows a poisson distribution. However, we learn the parameter,  $\lambda$  that governs the poisson distribution from real data.

## 3 Motivation: Bike Sharing

In this section, we formally describe a bike sharing system. It is compactly described using the following tuple:

$$\langle \mathcal{S}, \mathcal{V}, \mathbf{C}^\#, \mathbf{C}^*, \mathbf{d}^{\#,0}, \mathbf{d}^{*,0}, \{\sigma_v^0\}, \mathbf{F}, \mathbf{R}, \mathbf{P} \rangle$$

$\mathcal{S}$  represents the set of base stations.  $\mathcal{V}$  represents the set of carrier vehicles that can be employed to redeploy bikes.  $\mathbf{C}^\#$  is a vector representing docking capacity of all stations, with

$C_s^\#$  representing docking capacity of a station  $s \in \mathcal{S}$ . Similarly,  $C^*$  is a vector representing storage capacity of different carriers. Note that we assumed a fixed set of carriers in our model. However, in deciding the optimal number of carriers, our optimization approach can be run multiple times with different numbers of carriers to perform sensitivity analysis.

Distribution of bikes at a base station  $s$  at any time  $t$  is given by  $d_s^{\#,t}$ . Hence, initial distribution at any station  $s$  (provided as input) is denoted by  $d_s^{\#,0}$ . Similarly, total number of bikes present in a carrier  $v$  at any time  $t$  is given by  $d_v^{*,t}$  while the initial allotment of bikes  $d_v^{*,0}$  is provided as input.  $\sigma_v^0(s)$  captures the initial distribution of a carrier and is set to 1 if carrier  $v$  is stationed at station  $s$  initially<sup>1</sup>. For ease of notation in the optimization formulation, we use the generic  $\sigma_v^t(s)$  and set it to 0, if  $t > 0$ .

$F_{s,s'}^{t,k}$  represents the expected demand at time step  $t$  going out from station  $s$  and reaching station  $s'$  after  $k$  time steps. For instance, to compute a redeployment and routing strategy for Mondays, this would be computed by averaging demand over all the Mondays.  $R_{s,s'}^{t,k}$  represents the revenue obtained by the company if a bike is hired at time  $t$  from station  $s$  and returned at station  $s'$  after  $k$  time steps.  $P_{s,s'}$  represents the penalty for any carrier vehicle to travel from  $s$  to  $s'$ .

Given the customer demand of bikes between different stations, the goal is to maximize profit<sup>2</sup> or minimize loss of the bike sharing company by redeploying bikes using carrier vehicles (to satisfy customer demand). Notice that, minimizing lost demand would also minimize number of stations that are empty or full. As indicated earlier, we refer to this as the Dynamic Redeployment and Routing Problem (DRRP).

**Proposition 1** *Solving DRRP is an NP-Hard problem.*

**Proof Sketch.** We show that DRRP is a generalisation of 3-set partitioning problem, a known NP-Hard problem. ■

## 4 Solution Approach

We employ a data driven approach to solve DRRP. That is to say, we compute redeployment and routing strategies for a given training data set of demand values and evaluate the performance of the computed redeployment and routing strategies on a testing data set. Specifically, we provide a Mixed Integer Linear Problem (MILP) formulation for solving DRRP with expected demand values,  $\mathbf{F}$  that are obtained from the training data set. For ease of understanding, the decision and intermediate variables employed in the formulation are provided in Table 1.

We have access to flow of bikes in  $\mathbf{F}$ . Note that we only have the information about successful bike trips, thus we employ a standard method adopted in (Shu et al. 2013) to predict the actual demand, where the demand is represented

<sup>1</sup>While we provide a generalized model for initial location of carriers, our model can easily be extended such that carriers are enforced to start and return in a particular depot.

<sup>2</sup>We do not directly minimize lost demand, because that can result in a significant cost due to carrier vehicles. Profit provides the right trade-off between minimizing lost demand (maximizing revenue) and reducing cost due to carriers.

Category	Variable	Definition
Decision	$y_{s,v}^{+,t}$	Number of bikes picked from $s$ by carrier $v$ at time $t$
	$y_{s,v}^{-,t}$	Number of bikes dropped at $s$ by carrier $v$ at time $t$
	$z_{s,s',v}^t$	Set to 1 if carrier $v$ has to move from $s$ to $s'$ at time $t$
Intermediate	$x_{s,s'}^{t,k}$	Number of bikes moving from $s$ at time $t$ to $s'$ at $t+k$
	$d_s^{\#,t}$	Number of bikes present in station $s$ at time-step $t$
	$d_v^{*,t}$	Number of bikes present in carrier $v$ at time $t$

Table 1: Decision and Intermediate Variables

$$\begin{aligned} \min_{y^+, y^-, z} & - \sum_{t,k,s,s'} R_{s,s'}^{t,k} \cdot x_{s,s'}^{t,k} + \sum_{t,v,s,s'} P_{s,s'} \cdot z_{s,s',v}^t \quad (1) \\ \text{s.t. } & d_s^{\#,t} + \sum_{k,\hat{s}} x_{\hat{s},s}^{t-k,k} - \sum_{k,s'} x_{s,s'}^{t,k} + \\ & \sum_v (y_{s,v}^{-,t} - y_{s,v}^{+,t}) = d_s^{\#,t+1}, \quad \forall t, s \quad (2) \\ & x_{s,s'}^{t,k} \leq d_s^{\#,t} \cdot \frac{F_{s,s'}^{t,k}}{\sum_{k,\hat{s}} F_{s,\hat{s}}^{t,k}}, \quad \forall t, k, s, s' \quad (3) \\ & d_v^{*,t} + \sum_{s \in \mathcal{S}} [(y_{s,v}^{+,t} - y_{s,v}^{-,t})] = d_v^{*,t+1}, \quad \forall t, v \quad (4) \\ & \sum_{k \in \mathcal{S}} z_{s,k,v}^t - \sum_{k \in \mathcal{S}} z_{k,s,v}^{t-1} = \sigma_v^t(s), \quad \forall t, s, v \quad (5) \\ & \sum_{j \in \mathcal{S}, v \in \mathcal{V}} z_{s,j,v}^t \leq 1, \quad \forall t, s \quad (6) \\ & y_{s,v}^{+,t} + y_{s,v}^{-,t} \leq C_v^* \cdot \sum_{i \in \mathcal{S}} z_{s,i,v}^t, \quad \forall t, s, v \quad (7) \\ & 0 \leq x_{s,s'}^{t,k} \leq F_{s,s'}^{t,k}, 0 \leq d_s^{\#,t} \leq C_s^\#, 0 \leq y_{s,v}^{+,t} \leq C_v^*, \\ & 0 \leq y_{s,v}^{-,t} \leq C_v^*, 0 \leq d_v^{*,t} \leq C_v^* \quad (8) \\ & z_{i,j,v}^t \in \{0, 1\} \quad (9) \end{aligned}$$

Table 2: SOLVEDRRP()

using a *Poisson distribution* with mean computed from historical data. One of our goals is to compute a redeployment of bikes and it should be noted that because of redeployment, the number of bikes at a station will be different to what was observed in the training dataset. Hence, flow of bikes between station  $s$  and  $s'$  at time step  $t$  for  $k$  time steps suggested by our MILP will be different from the observed flow of bikes in the data, i.e.,  $F_{s,s'}^{t,k}$ . To represent this, we introduce a proxy variable,  $x_{s,s'}^{t,k}$  for  $F_{s,s'}^{t,k}$  that is set based on  $\mathbf{F}$  and the number of bikes available in the source station after redeployment.  $\mathbf{x}$  is included in the objective to ensure most of the expected demand is satisfied. For this reason,  $\mathbf{x}$  is only an intermediate variable that is proxy to expected demand,  $\mathbf{F}$ . The *lost demand* is calculated as the difference between demand sample ( $\mathbf{F}$ ) and the served demand ( $\mathbf{x}$ ).

To represent the trade-off between lost demand (or equivalently revenue from bike jobs) and cost of using carrier vehicles accurately, we employ the dollar value of both quanti-

ties and combine them into overall profit<sup>3</sup>. This objective is represented in Equation (1) of the MILP in SOLVEDRRP().

The key to this formulation for solving DRRP are the flow preservation, movement and capacity constraints for bikes, stations and carriers. Here, we provide a brief description of these constraints:

**1. Flow of bikes in and out of stations is preserved:** Constraints (2) enforces this flow preservation by ensuring equivalence of the number of bikes at station,  $s$  at time,  $t+1$  (i.e.,  $d_s^{\#,t+1}$ ) to the sum of bikes at station,  $s$  at time,  $t$  (i.e.,  $d_s^{\#,t}$ ) and net number of bikes arriving at station  $s$  after time step  $t$  (i.e.,  $\sum_{k,\hat{s}} x_{\hat{s},s}^{t-k,k} - \sum_{k,s'} x_{s,s'}^{t,k}$ ), minus the net number of bikes picked up from station  $s$  by all carriers  $v$  (i.e.,  $\sum_v [y_{s,v}^+,t - y_{s,v}^-,t]$ ).

**2. Flow of bikes between any two stations follows the transition dynamics observed in the data:** As a subset of arrival customers can be served if number of bikes present in the station is less than arrival demand, constraints (3) ensures that flow of bikes between any two station  $s$  and  $s'$  should be less than the product of number of bikes present in the source station  $s$  (i.e.  $d_s^{\#,t}$ ) and the transition probability that a bike will move from  $s$  to  $s'$  (according to customer demand (i.e.  $F_{s,s'}^{t,k} / \sum_{k,\hat{s}} F_{s,\hat{s}}^{t,k}$ )).

**3. Flow of bikes in and out of carriers is preserved:** Constraints (4) enforces this flow is preserved by ensuring equivalence of the number of bikes in a carrier,  $v$  at time step  $t+1$  (i.e.,  $d_v^{*,t+1}$ ) to the sum of bikes in carrier,  $v$  at time step  $t$  (i.e.,  $d_v^{*,t}$ ) and the net number bikes added at time step  $t$  to carrier  $v$  (i.e.,  $\sum_{s \in S} [y_{s,v}^+,t - y_{s,v}^-,t]$ ).

**4. Flow of carriers in and out of stations is preserved<sup>4</sup>:** Since  $\sigma_v^t = 0$  for all  $t > 0$ , constraints (5) ensures that flow out of a station  $s$  for a carrier  $v$  at time  $t$  (i.e.,  $\sum_{k \in S} z_{s,k,v}^t$ ) is equivalent to flow of  $v$  into the station  $s$  at time  $t-1$  (i.e.,  $\sum_{k \in S} z_{k,s,v}^{t-1}$ ). For  $t = 0$ , depending on  $\sigma_v^0$  is given as input, this constraint will ensure carrier flow moves appropriately out of the initial locations.

**5. Only one carrier can be in one station at a time step:** Constraints (6) ensures this by restricting the maximum carrier flow in a station as one.

**6. Carrier can pick up or drop off bikes from a station by being at the station:** Constraints (7) enforces that the number of bikes picked up or dropped off at a time is bounded by whether the station is visited at that time step.

**7. Station capacity is not exceeded when redeploying bikes:** Constraints (8) ensures that the number of bikes at a station,  $s$  is lower than the number of docks available at that station (i.e.,  $C_s^{\#}$ ).

**8. Carrier capacity is not exceeded when redeploying bikes:** Constraints (9) ensures that the number of bikes dropped off or picked up from any station at every time step and in aggregation is always less than the carrier capacity.

<sup>3</sup>We do not consider the labor/capital cost in the optimization model as they are constant (decided by strategic planning) and would not alter the results. However, we have a buffer on fuel cost to account for any other costs pertaining to day-to-day operations.

<sup>4</sup>Note that this constraint does not preclude a carrier from staying in the same station.

## 5 Decomposition Approach for Solving DRRP

We now provide a decomposition approach to exploit the minimal dependency that exists in the MILP of SOLVEDRRP() between the routing problem (how to move carrier vehicles between base stations to pick up or drop off bikes) and the redeployment problem (how many bikes and from where to pick up and drop off bikes). The following observation highlights this minimal dependency:

**Observation 1** *In the MILP of Table 2:*

- $y^+$  and  $y^-$  variables capture the solution for the redeployment problem.

- $z$  variables capture the solution for the routing problem.

*These sets of variables only interact due to constraints (7). In all other constraints of the optimization problem, the routing and redeployment problems are completely independent.*

In order to exploit Observation 1, we use the well known Lagrangian Dual Decomposition (Fisher 1985; Gordon et al. 2012) technique. While this is a general purpose approach, its scalability, usability and utility depend significantly on **whether the right constraints are dualized<sup>5</sup> and if primal solution can be extracted from an infeasible dual solution<sup>6</sup>**. If the right constraints are not dualized or primal solution is not extracted, then the underlying Lagrangian based optimization may not provide desired scalability and solution quality. In order to provide a sense of the overall flow, the pseudo code for LDD is provided in Algorithm 1. We first identify the decomposition of the optimization problem into a master problem and slaves (SOLVEREDEPLOY() and SOLVEROUTING()). As highlighted in observation 1, only constraints (7) contains a dependency between routing and redeployment problems. Thus, we dualize constraints (7) using the price variables,  $\alpha_{s,t,v}$  and obtain the Lagrangian as follows:

$$\begin{aligned} \mathcal{L}(\alpha) &= \min_{z,y^+,y^-} \left[ - \sum_{t,k,s,s'} R_{s,s'}^{t,k} \cdot x_{s,s'}^{t,k} + \sum_{t,v,s,s'} P_{s,s'} \cdot z_{s,s',v}^t \right. \\ &\quad \left. + \sum_{s,t,v} \alpha_{s,t,v} \cdot (y_{s,v}^+,t + y_{s,v}^-,t - C_v^* \cdot \sum_i z_{s,i,v}^t) \right] \quad (10) \\ &= \min_{y^+,y^-} \left[ - \sum_{t,k,s,s'} R_{s,s'}^{t,k} \cdot x_{s,s'}^{t,k} + \sum_{s,t,v} \alpha_{s,t,v} \cdot (y_{s,v}^+,t + y_{s,v}^-,t) \right] \\ &\quad + \min_z \left[ \sum_{t,v,s,s'} P_{s,s'} \cdot z_{s,s',v}^t - \sum_{s,t,v} \alpha_{s,t,v} \cdot C_v^* \cdot \sum_i z_{s,i,v}^t \right] \quad (11) \end{aligned}$$

In Equation (11), the first two terms correspond to the redeployment problem and the second two terms correspond to the routing problem. Thus, we have a decomposition of the dual problem into two slaves. Specifically, the slave optimization formulations corresponding to the redeployment and routing problems after decomposition are given in Table 3 and Table 4 respectively.

To obtain the final solution for the original optimization problem of Table 2, at the master, we optimize  $\max_{\alpha} \mathcal{L}(\alpha)$ .

<sup>5</sup>So that resulting subproblems are easy to solve and the upper bound derived from the LDD approach is tight.

<sup>6</sup>So that we can derive a valid lower bound (heuristic solution) during LDD process.

$$\min_{\mathbf{y}^+, \mathbf{y}^-} - \sum_{t,k,s,s'} R_{s,s'}^{t,k} \cdot x_{s,s'}^{t,k} + \sum_{s,t,v} \alpha_{s,v}^t \cdot (y_{s,v}^{+,t} + y_{s,v}^{-,t})$$

**s.t.** Constraints 2, 3, 4 & 8 hold

Table 3: SOLVEREDEPLOY()

$$\min_{\mathbf{z}} \sum_{t,v,s,s'} P_{s,s'} \cdot z_{s,s',v}^t - \sum_{s,t,v} \alpha_{s,t,v} \cdot C_v^* \cdot \sum_i z_{s,i,v}^t$$

**s.t.** Constraints 5, 6 & 9 hold

Table 4: SOLVEROUTING()

From Equation (11), given an  $\alpha$ , the dual value corresponding to the original problem is obtained by adding up the solution values from the two slaves. This **master** optimization problem is solved iteratively using sub-gradient descent on price variables,  $\alpha$ :

$$\alpha_{s,t,v}^{k+1} = [\alpha_{s,t,v}^k + \gamma \cdot (y_{s,v}^{+,t} + y_{s,v}^{-,t} - C_v^* \cdot \sum_i z_{s,i,v}^t)]_+ \quad (12)$$

where  $[\cdot]_+$  notation indicates that if the value within square brackets is less than 0, then we consider it as zero and if it is positive, we take that value as it is. This is so, because we have dualized a less than equal to constraint and a value of less than zero indicates there is no violation of the constraint.  $\gamma$  corresponds to step parameter that is derived using standard strategies highlighted in (Bertsekas 1999, Section 6.3.1). The value within parenthesis ( ) in Equation (12) is computed from the solutions of the two slaves.

Convergence in the process is detected when the difference between the primal objective (defined as  $p$  in Algorithm 1) and the dual objective (the sum of the slave's objectives  $o_1, o_2$ ) is less than a pre-determined threshold value. In order to determine convergence of the algorithm and also understand the progress towards computing the optimal solution, we need the best primal solution in conjunction with the dual solution. Therefore, extracting the best primal solution after each iteration of solutions from slaves is critical. This is also challenging because the solution obtained from slaves may not always be feasible for the original problem in Table 2.

---

**Algorithm 1: SolveLDD(*drrp*)**

---

**Initialize:**  $\alpha^0, it \leftarrow 0$ ;

**repeat**

$o_1, \mathbf{x}, \mathbf{y}^- \leftarrow \text{SOLVEREDEPLOY}(\alpha^{it}, drrp)$   
 $o_2, \mathbf{z} \leftarrow \text{SOLVEROUTING}(\alpha^{it}, drrp)$   
 $\alpha_{s,v}^{it+1,t} \leftarrow [\alpha_{s,v}^{it,t} + \gamma \cdot (y_{s,v}^{+,t} + y_{s,v}^{-,t} - C_v^* \cdot \sum_i z_{s,i,v}^t)]_+$   
 $p, x_p, y_p^-, y_p^+ \leftarrow \text{EXTRACTPRIMAL}(\mathbf{z}, drrp)$ ;  
 $it \leftarrow it + 1$ ;

**until**  $[p - (o_1 + o_2)] \leq \delta$ ;

**return**  $p, x_p, y_p^+, y_p^-, z$

---

**Observation 2** *The infeasibility in the dual solution arises because routes of the carriers (computed by routing slave)*

*may not be consistent with redeployment of bikes (computed by redeployment slave). However, solution of the routing slave is always feasible and can be fixed in the optimization problem of Table 2 to obtain a feasible primal solution.*

Let  $Z_{s,v}^t = \sum_{s'} z_{s,s',v}^t$ . We extract the primal solution by solving the following optimization problem provided in Table 5 and subtract the routing cost from the objective to get the primal solution.

$$\max_{\mathbf{y}^+, \mathbf{y}^-} \sum_{t,k,s,s'} R_{s,s'}^{t,k} \cdot x_{s,s'}^{t,k}$$

**s.t.** Constraints 2, 3, 4, 8 hold and

$$y_{s,v}^{+,t} + y_{s,v}^{-,t} \leq C_v^* \cdot Z_{s,v}^t, \forall t, s, v \quad (13)$$

Table 5: EXTRACTPRIMAL()

**Proposition 2** (Fisher 1985) *Error in solution quality obtained by Lagrangian dual decomposition method in Algorithm 1 is bounded by  $p - (o_1 + o_2)$ .*

It should be noted that MILP in Table 2 provides the optimal solution.

## 6 Abstraction Approach for Solving DRRP

Even after applying LDD, we can only scale to problems with at most 60 stations, 38 time-step and 6 carriers. However, in bigger cities, the number of base stations is in the order of hundreds. To ensure scalability to bigger cities, we propose a heuristic approach that employs abstraction. Specifically, we have the following key steps:

- Create an abstract DRRP with abstract stations, each of which is a grouping of original base stations.
- Solve the abstract DRRP using LDD and obtain routing and redeployment strategy over abstract stations.
- Derive the routing and redeployment strategies for the original DRRP from the routing and redeployment strategies for abstract DRRP.

### Create abstract DRRP

Concretely, the first step in this approach is to generate the abstract DRRP,  $\langle \tilde{\mathcal{S}}, \mathcal{V}, \tilde{\mathbf{C}}^\#, \mathbf{C}^*, \tilde{\mathbf{d}}^{\#,0}, \mathbf{d}^{*,0}, \{\tilde{\sigma}_v^0\}, \tilde{\mathbf{F}}, \tilde{\mathbf{R}}, \tilde{\mathbf{P}} \rangle$  from the original DRRP. Everything related to carriers in the abstract DRRP remains the same as in the original DRRP. In practice, revenue,  $R_{s,s'}^{t,k}$  is only dependent on the time of the day,  $t$  and number of time steps,  $k$  for which the the bike is hired and not on  $s$  and  $s'$ . Hence, we can assume that the revenue model also remains the same in the abstract DRRP. We outline below how the other elements of the abstracted DRRP tuple are computed from the original DRRP:

- Stations in abstracted DRRP,  $\tilde{\mathcal{S}}$ : Grouping of stations,  $\mathcal{S}$  into abstract stations can either be provided by an expert or done manually or computed through clustering approaches (ex: k-means clustering). Thus, each abstracted station,  $\tilde{s}$  is a set of original base stations.
- Capacity of an abstract station,  $C_{\tilde{s}}^\# = \sum_{s \in \tilde{s}} C_s^\#$ . The capacity of a abstract station  $\tilde{s}$  is the sum of capacities of all the stations  $s \in \tilde{s}$ .

- Initial distribution of the abstracted station:  $d_{\tilde{s}}^{\#,0} = \sum_{s \in \tilde{s}} d_s^{\#,0}$ . Initial distribution of a abstract station  $\tilde{s}$  is the sum of initial distribution of all the stations  $s \in \tilde{s}$ .
- Initial distribution of carrier :  $\sigma_{v,\tilde{s}}^0 = 1$ , if  $\exists s \in \tilde{s}$  and  $\sigma_{v,s}^0 = 1$ . That is to say, the carrier  $v$  is initially located in abstract station  $\tilde{s}$  if its original location (station  $s$ ) belongs to the abstract station  $\tilde{s}$
- Flow in abstracted DRRP:  $F_{\tilde{s},\tilde{s}'}^{t,k} = \sum_{\{s \in \tilde{s}, s' \in \tilde{s}'\}} F_{s,s'}^{t,k}$ . Flow from abstracted station  $\tilde{s}$  to  $\tilde{s}'$  is calculated as the sum of flows between any station  $s \in \tilde{s}$  to  $s' \in \tilde{s}'$  in the original DRRP.
- Cost model for the carriers in abstract DRRP:  $P_{\tilde{s},\tilde{s}'} = \max_{\{s \in \tilde{s}, s' \in \tilde{s}'\}} P_{s,s'}$ . While there are multiple models possible, we consider the conservative option of taking the worst case penalty. Specifically, we take the maximum penalty for traveling between any pair of stations  $s \in \tilde{s}$  and  $s' \in \tilde{s}'$ .

### Solve abstract DRRP

We use LDD from previous section to solve the abstract DRRP. There are two key outputs: (a) Redeployment strategy,  $\tilde{y}$  for moving bikes between abstract stations; and (b) Routing strategy,  $\tilde{z}$  for moving carriers between abstract stations,  $\tilde{s}$  at different time steps.

$$\begin{aligned}
& \max_{y^+, y^-, z} \sum_{t,k,s,s'} R_{s,s'}^{t,k} \cdot x_{s,s'}^{t,k} - \sum_{t,v,s,s'} P_{s,s'} \cdot z_{s,s',v}^t \\
& \text{s.t. Constraints 2-9 hold and} \\
& \sum_{s \in \tilde{s}, s' \in \tilde{s}'} z_{s,s',v}^t = \tilde{z}_{\tilde{s},\tilde{s}',v}^t \quad \forall \tilde{s}, \tilde{s}', t, v \quad (14)
\end{aligned}$$

Table 6: **RETRIEVEDRRP()**

### Derive strategies for original DRRP

We retrieve the solution of DRRP from solutions obtained for abstract DRRP.  $\tilde{z}$  is the routing strategy obtained by solving the abstract DRRP, where  $\tilde{z}_{\tilde{s},v}^t = \sum_{\tilde{s}'} \tilde{z}_{\tilde{s},\tilde{s}',v}^t = 1$  entails carrier  $v$  is present in abstract station  $\tilde{s}$  at time step  $t$ . Table 6 provides the optimization model to retrieve the solution for DRRP with the assumption of carrier travels one station in each time step. We solve the global MIP `SolveDRRP()` provided in Table 2 with an additional set of constraints (14) that ensure a carrier can only be present in a base station at any time step if the station belongs to the abstract station where the carrier is located in the abstract DRRP solution. It also enforces that the decision variable  $z_{s,s',v}^t$  can only be 1 if  $s \in \tilde{s}$ ,  $s' \in \tilde{s}'$  and  $\tilde{z}_{\tilde{s},\tilde{s}',v}^t = 1$ . In the MILP of `RetrieveDRRP()`, we explicitly set the decision variables  $z_{s,s',v}^t$  to 0 if  $s \in \tilde{s}$ ,  $s' \in \tilde{s}'$  and  $\tilde{z}_{\tilde{s},\tilde{s}',v}^t = 0$ . Thus `RetrieveDRRP()` becomes an easier optimization problem than `SolveDRRP()`.

## 6.1 Discussion

As we are abstracting the base stations based on their relative distance, all the base stations within an abstract station are located nearby. So, in reality it is possible for a carrier to visit all the base stations of an abstract station within one time-step. In this case, the mechanism to retrieve the DRRP

solution from abstract station level solution changes and is outlined here.

$$\begin{aligned}
& \max_{y^+, y^-} \sum_{t,s,s'} R_{s,s'}^{t,k} \cdot x_{s,s'}^{t,k} \quad (15) \\
& \text{s.t. } d_{\tilde{s}}^{\#,t} + \sum_{k,\tilde{s}} x_{\tilde{s},\tilde{s}}^{t-k,k} - \sum_{k,\tilde{s}'} x_{\tilde{s},\tilde{s}'}^{t,k} \\
& \quad y_s^{-,t} - y_s^{+,t} = d_s^{\#,t+1} \quad , \forall t, s \quad (16) \\
& x_{s,s'}^{t,k} \leq d_s^{\#,t} \cdot \frac{F_{s,s'}^{t,k}}{\sum_{k,\tilde{s}} F_{s,\tilde{s}}^{t,k}} \quad , \forall t, k, s, s' \quad (17) \\
& y_s^{+,t} + y_s^{-,t} \leq C_v^* \cdot Z_s^t \quad , \forall t, s \quad (18) \\
& \sum_{s \in \tilde{s}} [y_s^{+,t} - y_s^{-,t}] = d_v^{*,t+1} - d_v^{*,t} \quad , \forall t, \tilde{s} \quad (19) \\
& 0 \leq x_{s,s'}^{t,k} \leq F_{s,s'}^{t,k}, 0 \leq y_s^{+,t}, y_s^{-,t} \leq C_v^*, 0 \leq d_s^{\#,t} \leq C_s^{\#} \quad (20)
\end{aligned}$$

Table 7: **GETSTATIONREDEPLOY( $v, \mathbf{Z}, \mathbf{d}_v^*$ )**

We first compute redeployment strategy,  $z$  at the level of base stations for each carrier over the entire horizon and then compute the routing strategy within each abstract station. The optimization problem of Table 7 employs the constants,  $\mathbf{Z}$  to obtain a base station level redeployment strategy,  $\mathbf{y}$ . If  $s$  is an original station and  $\tilde{s}$  is an abstract station, then let  $Z_s^t = 1$ , if  $s \in \tilde{s}$  and  $\sum_v \tilde{z}_{\tilde{s},v}^t = 1$ . Our objective function specified in (15) is to maximize the overall revenue of the agency that implies maximum number of customers are served. One of the key differentiating constraints that has not been used earlier is constraints (19). This ensures that total number of bikes picked up or dropped off from all base stations in an abstract station is equal to the number of bikes picked up or dropped off in the abstract station level redeployment strategy.

$$\begin{aligned}
& \min_{z} \sum_{t,s,s'} P_{s,s'} \cdot z_{s,s'}^t \quad (21) \\
& \text{s.t. } \hat{d}^{*,t} + \sum_s (Y_s^+ - Y_s^-) \cdot \sum_{s'} z_{s,s'}^t = \hat{d}^{*,t+1}, \forall t \in \hat{T} \quad (22) \\
& \sum_{t,s'} z_{s,s'}^t = 1 \quad , \forall s \in \tilde{s} | (Y_s^+ + Y_s^-) > 0 \quad (23) \\
& \sum_{s'} z_{s,s'}^t - \sum_{\tilde{s}} z_{\tilde{s},s}^{t-1} = \sigma^t(s) \quad , \forall t \in \hat{T}, s \in \tilde{s} \quad (24)
\end{aligned}$$

Table 8: **GETINTRAROUTING( $\tilde{s}, \mathbf{Y}$ )**

Given the base station level redeployment strategy,  $\mathbf{Y}$ , we now compute the best route within the stations of an abstract station,  $\tilde{s}$ . This problem can be solved locally for each abstract station,  $\tilde{s}$ , where carrier  $v$  is redeploying at time step  $t$ . If we have  $|\hat{T}|$  time-steps and  $|\mathcal{V}|$  carriers, we can solve  $|\hat{T}| \cdot |\mathcal{V}|$  subproblems separately. To figure out the initial location, we find a station within the abstract station which is nearest to the station from where the carrier has exited in the previous time-step. Since the position of carriers is

known at the first time step, we know the starting location. Such an approach automatically minimizes the inter-cluster routing. Table: 8 provides the optimization formulation to solve each subproblem. The objective specified in (21) is to minimize the routing cost. Constraints (22) ensure the re-deployment number from each station ( $\mathbf{Y}$ ) is satisfied and Constraints (23) ensure that each base station where a re-deployment is required is visited only once.

## 7 Experimental Settings

We evaluate our approaches with respect to run-time, revenue for company and lost demand on real world<sup>7</sup> and synthetic data sets. These data sets contain the following data: (1) Customer trip records that are indicative of successful bookings. We predict demand from these trip records. (2) Number of active docks in each station (i.e. station capacity) and initial distribution of bikes in the station at the beginning of a day. (3) Geographical locations of base stations. From the longitude and latitude information of stations, we calculate the relative distance between two stations. (4) Revenue model of the agency<sup>8</sup>. (5) Cost of fuel for carriers<sup>9</sup>.

We generated our synthetic data set as follows: (a) We take a subset of the stations from the real world data set (b) Customer demands, station capacity, geographical location of stations and initial distribution are drawn from the real world data for those specific stations. (c) We take the same revenue and cost model discussed earlier from real datasets. Because of limited scalability of MILP and LDD without abstraction, we are only able to evaluate run-time performance on small scale synthetic problems.

To the best of our knowledge, there is no other approach that addresses this problem nor does there exist an approach that can easily be adapted to solve our problem. Hence we compare our approaches against current practice of re-deploying at the end of the day (in which user activities during the rebalancing period are negligible) with respect to: (a) overall revenue generated for the agency; and (b) lost demand. We compute the outcome of current practice by simulating flow of bikes between stations to be proportional to the flow observed in the data. With the flow, we compute distribution of bikes and based on that, we can compute the revenue and loss in demand.

### Synthetic Dataset——

We have three sets of results<sup>10</sup> on the synthetic data set.

<sup>7</sup>Data is from two leading US bike sharing companies: *CapitalBikeShare* [<http://capitalbikeshare.com/system-data>] and *Hubway* [<http://hubwaydatachallenge.org/trip-history-data>]

<sup>8</sup>Typically, first 30 minutes for subscription rides is free. After that money is charged. In our model, to ensure consistency, we can represent revenue for first 30 minutes as the subscription fees divided by the average number of rides.

<sup>9</sup>Mileage results are shown in Table 2 of (Fishman, Washington, and Haworth 2014) and [http://www.globalpetrolprices.com/diesel\\_prices/#USA](http://www.globalpetrolprices.com/diesel_prices/#USA), shows that diesel price is 1.01 USD per liter, but we assumed it as 1.5 USD to consider worst case.

<sup>10</sup>All the linear optimization models were solved using IBM ILOG CPLEX Optimization Studio V12.5 incorporated within

Firstly, we compare the runtime performance of LDD (SOLVE<sub>LDD</sub>()) with the global MILP (SOLVE<sub>MILP</sub>()) in Figure 2b. X-axis denotes the scale of the problem where we varied the number of stations from 5 to 50. Y-axis denotes the total time taken in seconds on a logarithmic scale. LDD significantly outperforms global MILP once the number of stations is above 10. Specifically, global MILP was unable to finish within a cut-off time of 6 hours for any problem with more than 20 stations, while LDD was able to solve problems with 50 stations in an hour.

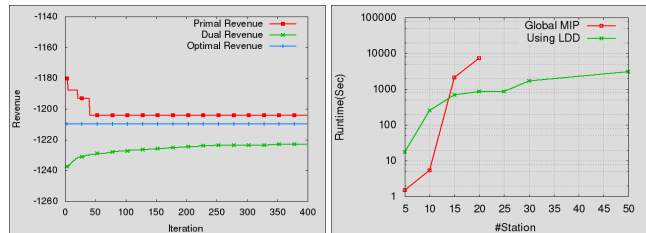


Figure 2: (a) Duality gap (b) Runtime: LDD vs Global MILP

In the second set of results we demonstrate the convergence of LDD. LDD can achieve the optimal solution if the duality gap i.e. the gap between primal and dual solution becomes zero. Figure: 2a shows that the duality gap for a 20 station problem is only 1%. While, we do not show the results here, on larger problems we are able to get a solution with duality gap of less than 0.5 %.

Finally, we demonstrate the performance of abstraction in comparison with optimal on a problem with 30 base stations. We grouped those 30 base stations into 8 abstract stations. Then we run the LDD based optimization on both the base station and abstraction station problems. Table: 9 shows the effect of abstraction on the generated revenue and execution time based on five random instances of customer demand. While there is a very small drop in revenue (0.2%) due to abstraction, we get a very significant (at least an order of magnitude gain) in runtime.

Instance	With Abstraction		Without Abstraction	
	Revenue	Runtime (sec)	Revenue	Runtime (sec)
1	23580	51	23640	3840
2	23627	106	23678	3540
3	23610	57	23727	3120
4	23613	49	23645	3150
5	23519	45	23590	3119

Table 9: Effect of Abstraction

### Real-world Datasets——

Majority of our results are provided on the *CapitalBikeShare* dataset. This data set has 305 active stations and we consider 50 abstract stations (obtained through k-means clustering). The planning horizon is 38 ( 30 minute intervals during the working hours from 5AM-12AM).

python code on a 3.2 GHz Intel Core i5 machine with 4GB DDR3 RAM

We now provide the performance comparison between our approaches and current practice (i.e., no redeployment during the day) with respect to lost demand and revenue generated for the bike-sharing company. We generate the overall mean demand as well as the demand for individual weekdays from historical data of trips. We compute the results for the entire time horizon 5 AM to 12 AM and also for one of the peak durations from 5 AM to 12 PM. Table:10 shows the percentage gain in revenue and the percentage reduction in lost demand in comparison with current practice. With respect to both revenue gain and lost demand, our approach (abstraction + LDD + MILP) was able to outperform current practice during the peak time as well as over the entire day. We reduce the lost demand in all the cases by at least 20%, a significant improvement over current practice.

	Whole day (5am-12am)		Peak period (5am-12pm)	
	Revenue gain	Lost demand reduction	Revenue gain	Lost demand reduction
Mean	3.47 %	22.72 %	7.74 %	30.58 %
Mon	2.33 %	22.46 %	4.48 %	25.55 %
Tue	3.07 %	28.56 %	7.86 %	37.10 %
Wed	3.30 %	31.16 %	8.95 %	44.88 %
Thu	2.86 %	33.76 %	6.04 %	35.97 %
Fri	2.51 %	27.37 %	4.50 %	28.15 %
Sat	3.87 %	23.61 %	4.33 %	24.30 %
Sun	3.01 %	26.00 %	4.04 %	36.51 %

Table 10: Revenue and lost demand comparison

The next set of results demonstrate the sensitivity of our approach with respect to small variations in demand. We created a set of 10 demands for each of the weekdays from the underlying *Poisson distribution* with mean calculated from the real world data set. For individual demand instances, we calculate the revenue and lost demand by applying our redeployment policy and compare it with the traditional policy. Figure 3 shows the mean and deviation of the revenue and lost call for each of the weekdays. Even considering the variance, Figure 3a shows that the revenue generated by following our redeployment strategy is still better (albeit by a small amount) than current practice. More importantly, Figure 3b demonstrates that we are able to significantly reduce the lost demand on all the cases.

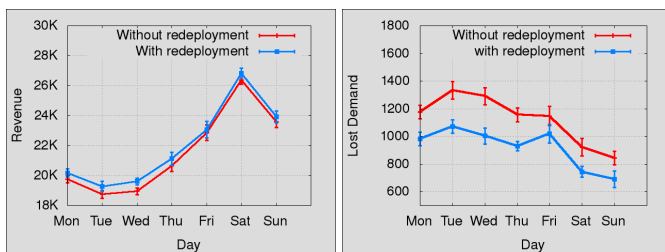


Figure 3: Sensitivity analysis: (a) Revenue comparison (b) Lost demand comparison

Next, we present the experimental results on the real world dataset of *Hubway*. *Hubway* BSS comprises with 95

active stations and we group them into 25 abstract stations. In the experiment we have used 4 carriers to redeploy the bikes. To predict the demands we have used 3-months of trip history records (3rd quarter of 2012). We took a planning horizon of 38 time-step (5 AM to 12 AM) in the experiment. Then we generate 7 instances of demand for each of the weekdays from the historical data.

We produce two sets of results with this data set. Table: 11 provides the comparison results (on revenue and lost demand) between the traditional approach and our dynamic redeployment approach. Our approach is able to gain an excess 5% in revenue on average while the lost demand is reduced by a minimum of 40 %.

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
Revenue Gain (%)	3.94	5.93	4.45	5.90	6.27	2.20	3.15
Lost Demand Reduction(%)	42.6	60.7	58.5	54.7	77.2	69.8	74.0

Table 11: Revenue and lost demand comparison (Hubway)

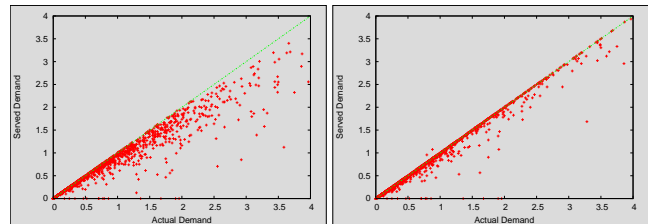


Figure 4: Correlation of demand and supply: (a) Without redeployment (b) With dynamic redeployment

Lastly, to visualize the effect of redeployment we draw the correlation between actual demand and served demand over the entire planning horizon. Since we aim to reduce the lost customer demand, it is better if most of the points are near line of equality or identity line. Figure: 4b illustrates the correlation between actual demand and demand served by following our redeployment model. Comparatively, with redeployment there are many more points closer to the identity line than with current practice (shown in Figure: 4a).

## 8 Conclusion

In this paper we addressed the dynamic redeploy problem in shared transportation systems. Our approach based on the Lagrangian Dual Decomposition and an abstraction based mechanism, addresses two key challenges (a) Provide an near-optimal policy for the dynamic redeployment of idle vehicles in conjunction with the routing solution for carriers (b) Provide a scalable solution for the real-world large scale problems. The empirical results on multiple real and synthetic data sets shown that our dynamic redeployment approach is not only able to achieve the original goal of reducing lost demand, but is also able to improve revenue for the bike sharing company, by using their existing resources. In future this work can be extended with a robust optimization technique which can account for all the realization of different demand scenarios.



## Acknowledgments

The research described in this paper was funded in part by the Singapore National Research Foundation (NRF) through the Singapore-MIT Alliance for Research and Technology (SMART) Center for Future Mobility (FM).

## References

- Bertsekas, D. P. 1999. *Nonlinear programming*. Athena Scientific, 2nd edition.
- Contardo, C.; Morency, C.; and Rousseau, L.-M. 2012. *Balancing a dynamic public bike-sharing system*, volume 4. CIRRELT.
- Fisher, M. L. 1985. An applications oriented guide to lagrangian relaxation. *Interfaces* 15(2):10–21.
- Fishman, E.; Washington, S.; and Haworth, N. L. 2014. Bike share's impact on car use: evidence from the united states, great britain, and australia. In *Proceedings of the 93rd Annual Meeting of the Transportation Research Board*.
- Fricker, C., and Gast, N. 2012. Incentives and redistribution in homogeneous bike-sharing systems with stations of finite capacity. *EURO Journal on Transportation and Logistics* 1–31.
- Gordon, G. J.; Varakantham, P.; Yeoh, W.; Lau, H. C.; Aravamudan, A. S.; and Cheng, S.-F. 2012. Lagrangian relaxation for large-scale multi-agent planning. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2012 IEEE/WIC/ACM International Conferences on*, volume 2, 494–501. IEEE.
- Lees-Miller, J. D.; Hammersley, J. C.; and Wilson, R. E. 2010. Theoretical maximum capacity as benchmark for empty vehicle redistribution in personal rapid transit. *Transportation Research Record: Journal of the Transportation Research Board* 2146(1):76–83.
- Leurent, F. 2012. Modelling a vehicle-sharing station as a dual waiting system: stochastic framework and stationary analysis. *Under review*, URL <https://hal.archives-ouvertes.fr/hal-00757228>.
- Meddin, R., and DeMaio, P. 2015. The bike-sharing world map. URL <http://www.bikesharingworld.com/>.
- Nair, R., and Miller-Hooks, E. 2011. Fleet management for vehicle sharing operations. *Transportation Science* 45(4):524–540.
- Pfrommer, J.; Warrington, J.; Schildbach, G.; and Morari, M. 2014. Dynamic vehicle redistribution and online price incentives in shared mobility systems. *Intelligent Transportation Systems, IEEE Transactions on* 15(4):1567–1578.
- Rainer-Harbach, M.; Papazek, P.; Hu, B.; and Raidl, G. R. 2013. *Balancing bicycle sharing systems: A variable neighborhood search approach*. Springer Berlin Heidelberg.
- 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.
- Saisubramanian, S.; Varakantham, P.; and Chuin, L. H. 2015. Risk based optimization for improving emergency medical systems. In *AAAI*.
- Schuijbroek, J.; Hampshire, R.; and van Hoes, W.-J. 2013. Inventory rebalancing and vehicle routing in bike sharing systems. *Working Paper, Carnegie Mellon University, Pittsburgh*.
- 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. *Technical report, 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.
- Yue, Y.; Marla, L.; and Krishnan, R. 2012. An efficient simulation-based approach to ambulance fleet allocation and dynamic redeployment. In *AAAI*.