Dynamic Redeployment to Counter Congestion or Starvation in Vehicle Sharing Systems

Supriyo Ghosh, Pradeep Varakantham, Yossiri Adulyasak[†], Patrick Jaillet[‡]
School of Information Systems, Singapore Management University
[†]Singapore-MIT Alliance for Research and Technology (SMART), Massachusetts Institute of Technology
[‡]Department of Electrical Engineering and Computer Science, Massachussets Institute of Technology
supriyog.2013@phdis.smu.edu.sg,pradeepv@smu.edu.sg,yossiri@smart.mit.edu,jaillet@mit.edu

Abstract

Vehicle-sharing (ex: bike sharing, car sharing) is widely adopted in many cities of the world due to concerns associated with extensive private vehicle usage, which has led to increased carbon emissions, traffic congestion and usage of non-renewable resources. In vehicle-sharing systems, base stations 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, typically, there is 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 of 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.

Introduction

Shared Transportation Systems (STS) offer the best 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 systems (ex: Car2go in Seattle, Zipcar in Pittsburgh), 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 [1].A bike-sharing system typically has a few hundred base stations scattered throughout a city. At the beginning of the day, each station is stocked with a predetermined 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. 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 carrier. We refer to the joint 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).

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 (as we later show in Proposition 1). Therefore, we focus on principled approximations and 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 hap-

[1]Meddin, R. and DeMaio, P., The bike-sharing world map. URL http://www.bikesharingworld.com/

pens) for each of the carriers.
(2) A method to decompose the overall optimization formulation into two components – one which computes redeployment strategy for bikes and one 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.

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 Hoeve 2013; Raviv and Kolka 2013; Raviv, Tzur, and Forma 2013; Raidl 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 actual routing cost for the carriers which is a major cost driver in BSS. 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. As can be observed from the data, customer demand of bikes varies over time stochastically and hence a myopic redeployment policy can significantly falter 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 and make sure that all the system wide demands should be served with a certain probability. 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) reports 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, λ that governs the poisson distribution from real data.

Motivation: Bike Sharing

In this section, we formally describe the specifics of a bike sharing system. A bike sharing system can be compactly described using the following tuple:

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

$\mathcal{S}$ represents the set of base stations and each station $s \in \mathcal{S}$ has a fixed capacity (number of docks) denoted by $C_s^{\#}$. $\mathcal{V}$ represents the set of carrier vehicles that can be employed to redeploy bikes and each carriers $v \in \mathcal{V}$ has a fixed capacity (number of bikes) denoted by C_v^*. 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. 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 actual demand at time step t going out from station s and reaching station s' after k time steps. $R_{s,s'}^{t,k}$ 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'.

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. ■

Optimization Model For Solving DRRP

We first provide a Mixed Integer Linear Problem (MILP) formulation for solving DRRP. 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 **F**. 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 $F_{s,s'}^{t,k}$

Category	Variable	Definition
Decision	$\hat{y}^t_{s,v}$	Number of bikes picked from s by carrier v at time t
	$\check{y}^t_{s,v}$	Number of bikes dropped at s by carrier v at time t
	$z^t_{s,s',v}$	Set to 1 if carrier v has to move from s to s' at time t)
Intermediate	$x^{t,k}_{s,s'}$	Number of bikes moving from s at time t to s' at $t+k$
	$d^{\#,t}_s$	Number of bikes present in station s at time-step t
	$d^{*,t}_v$	Number of bikes present in carrier v at time t

Table 1: **Decision and Intermediate Variables**

$$\min_{\mathbf{y},\mathbf{z}} - \sum_{t,k,s,s'} R^{t,k}_{s,s'} \cdot x^{t,k}_{s,s'} + \sum_{t,v,s,s'} P_{s,s'} \cdot z^t_{s,s',v} \tag{1}$$

$$\textbf{s.t. } d^{\#,t}_s + \sum_{k,\hat{s}} x^{t-k,k}_{\hat{s},s} - \sum_{k,s'} x^{t,k}_{s,s'} + \sum_v (\check{y}^t_{s,v} - \hat{y}^t_{s,v}) = d^{\#,t+1}_s, \quad \forall t, s \tag{2}$$

$$x^{t,k}_{s,s'} \leq d^{\#,t}_s \cdot \frac{F^{t,k}_{s,s'}}{\sum_{k,\hat{s}} F^{t,k}_{s,\hat{s}}}, \qquad \forall t, k, s, s' \tag{3}$$

$$d^{*,t}_v + \sum_{s\in S}[(\hat{y}^t_{s,v} - \check{y}^t_{s,v})] = d^{*,t+1}_v, \qquad \forall t, v \tag{4}$$

$$\sum_{k\in\mathcal{S}} z^t_{s,k,v} - \sum_{k\in\mathcal{S}} z^{t-1}_{k,s,v} = \sigma^t_v(s), \qquad \forall t, s, v \tag{5}$$

$$\sum_{j\in\mathcal{S}, v\in\mathcal{V}} z^t_{s,j,v} \leq 1, \qquad \forall t, s \tag{6}$$

$$\hat{y}^t_{s,v} + \check{y}^t_{s,v} \leq C^*_v \cdot \sum_i z^t_{s,i,v}, \qquad \forall t, s, v \tag{7}$$

$$0 \leq x^{t,k}_{s,s'} \leq F^{t,k}_{s,s'}, 0 \leq d^{\#,t}_s \leq C^{\#}_s, 0 \leq \hat{y}^t_{s,v}, \check{y}^t_{s,v} \leq C^*_v \tag{8}$$

$$0 \leq d^{*,t}_v \leq C^*_v, z^t_{i,j,v} \in \{0,1\} \tag{9}$$

Table 2: SOLVEDRRP()

that is set based on F and the number of bikes available in the source station after redeployment. **x** is included in the objective to ensure most of the expected demand is satisfied. For this reason, **x** is only an intermediate variable that is proxy to expected demand, **F**.

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 quantities and combine them into overall profit[2]. This objective is represented in Equation 1 of the MILP in SOLVEDRRP(). Intuitively, we have the following flow preservation, movement and capacity constraints for bikes, stations and carriers:

1. Flow of bikes in and out of stations is preserved: Constraints (2) enforce this flow preservation by ensuring equivalence of the number of bikes in and out of a station at each time step.

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) ensure 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 and the transition probability that a bike will move from s to s' .

3. Flow of bikes in and out of carriers is preserved: Constraints (4) enforce this flow is preserved by ensuring equivalence of the number of bikes in and out of a carrier at each time step.

4. Flow of carriers in and out of stations is preserved: Since $\sigma^t_v = 0$ for all $t > 0$, constraints (5) ensure that flow out of a station s for a carrier v at time t (i.e., $\sum_{k\in\mathcal{S}} z^t_{s,k,v}$) is equivalent to flow of v into the station s at time $t-1$ (i.e., $\sum_{k\in\mathcal{S}} z^{t-1}_{k,s,v}$). For $t = 0$, depending on σ^0_v 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) ensure 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) enforce 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) ensure 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) ensure 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.

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:*

- $\hat{\mathbf{y}}$ *and* $\check{\mathbf{y}}$ *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**[3] and **if primal**

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

[3]So that resulting subproblems are easy to solve and the upper bound derived from the LDD approach is tight

solution can be extracted from an infeasible dual solution[4].

Algorithm 1: SolveLDD($drrp$)

Initialize: $\alpha^0, it \leftarrow 0$;
repeat
  $o_1, \mathbf{x}, \check{\mathbf{y}}, \hat{\mathbf{y}} \leftarrow$ SOLVEREDEPLOY$(\alpha^{it}, drrp)$
  $o_2, \mathbf{z} \leftarrow$ SOLVEROUTING$(\alpha^{it}, drrp)$
  $\alpha^{it+1}_{s,t,v} \leftarrow \big[\alpha^{it}_{s,t,v} + \gamma \cdot (\hat{y}^t_{s,v} + \check{y}^t_{s,v} - C^*_v \cdot \sum_i z^t_{s,i,v})\big]_+$
  $p, x_p, \check{y}_p, \hat{y}_p \leftarrow$ EXTRACTPRIMAL $(\mathbf{z}, drrp)$;
  $it \leftarrow it + 1$;
until $\big[p - (o_1 + o_2)\big] \leq \delta$;
return $p, x_p, \hat{y}_p, \check{y}_p, z$

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:

$$\mathcal{L}(\alpha) = \min_{\mathbf{x},\mathbf{z},\hat{\mathbf{y}},\check{\mathbf{y}}} \Big[- \sum_{t,k,s,s'} R^{t,k}_{s,s'} \cdot x^{t,k}_{s,s'} + \sum_{t,v,s,s'} P_{s,s'} \cdot z^t_{s,s',v} + \sum_{s,t,v} \alpha_{s,t,v} \cdot (\hat{y}^t_{s,v} + \check{y}^t_{s,v} - C^*_v \cdot \sum_i z^t_{s,i,v}) \Big] \tag{10}$$

$$= \min_{\mathbf{x},\hat{\mathbf{y}},\check{\mathbf{y}}} \Big[- \sum_{t,k,s,s'} R^{t,k}_{s,s'} \cdot x^{t,k}_{s,s'} + \sum_{s,t,v} \alpha_{s,t,v} \cdot \big(\hat{y}^t_{s,v} + \check{y}^t_{s,v}\big) \Big] + \min_{\mathbf{z}} \Big[\sum_{t,v,s,s'} P_{s,s'} \cdot z^t_{s,s',v} - \sum_{s,t,v} \alpha_{s,t,v} \cdot C^*_v \cdot \sum_i z^t_{s,i,v} \Big] \tag{11}$$

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 nice decomposition of the dual problem into two slaves. More specifically, the **slave optimization** corresponding to the redeployment and routing problems are given in Table 3 and Table 4 respectively.

$$\min_{\hat{\mathbf{y}},\check{\mathbf{y}}} - \sum_{t,k,s,s'} R^{t,k}_{s,s'} \cdot x^{t,k}_{s,s'} + \sum_{s,t,v} \alpha_{s,t,v} \cdot (\hat{y}^t_{s,v} + \check{y}^t_{s,v})$$

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

Table 3: SOLVEREDEPLOY()

The dual value corresponding to the original problem can thus be obtained by adding up the solution values from the two slaves. It should be noted that we have only considered $\mathcal{L}(\alpha)$ so far. To obtain the final solution for the original optimization problem of Table 2, we have to solve the following optimization problem at the master in order to reduce violation of the dualized constraints: $\max_\alpha \mathcal{L}(\alpha)$. This **master** optimization problem is solved iteratively using subgradient descent on price variables, α:

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

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

Table 4: SOLVEROUTING()

$$\alpha^{k+1}_{s,t,v} = \big[\alpha^k_{s,t,v} + \gamma \cdot (\hat{y}^t_{s,v} + \check{y}^t_{s,v} - C^*_v \cdot \sum_i z^t_{s,i,v})\big]_+ \tag{12}$$

where $[]_+$ 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. γ corresponds to step parameter. The value within parenthesis () is computed from the solutions of the two slaves.

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.

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

$$\max_{\mathbf{y}} \sum_{t,k,s,s'} R^{t,k}_{s,s'} \cdot x^{t,k}_{s,s'}$$

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

$$\hat{y}^t_{s,v} + \check{y}^t_{s,v} \leq C^*_v \cdot Z^t_{s,v} \qquad ,\forall t,s,v \tag{13}$$

Table 5: EXTRACTPRIMAL()

Let $Z^t_{s,v} = \sum_{s'} z^t_{s,s',v}$. 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 objective.

Abstraction Approach for Solving DRRP

Even after applying LDD, we can only scale to problems with at most 60 base stations. However, in some of the bigger cities, the number of base stations is in the order of couple of hundreds. In order to provide scalable solutions for such problems, we propose a heuristic approach based on creating

[4]So that we can derive a valid lower bound (heuristic solution) during LDD process

abstract stations, each of which represents a set of original base stations. In this approach, we initially solve the abstract problem and then derive the solution to the original problem from the solution of the abstract problem.

Concretely, the first step in this approach is to generate the abstract DRRP, $\left\langle \tilde{\mathcal{S}}, \mathcal{V}, \tilde{\mathbf{C}}^{\#}, \mathbf{C}^{*}, \tilde{\mathbf{d}}^{\#,0}, \mathbf{d}^{*,0}, \{\tilde{\sigma}^{0}_{v}\}, \tilde{\mathbf{F}}, \tilde{\mathbf{R}}, \tilde{\mathbf{P}} \right\rangle$ from the original DRRP. Everything related to carriers remains the same as before, but the rest are computed from the original DRRP. In the second step, 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.

$$\max_{\mathbf{y}^{+},\mathbf{y}^{-}} \sum_{t,s,s,s'} R^{t,k}_{s,s'} \cdot x^{t,k}_{s,s'} \tag{14}$$

$$\textbf{s.t. } d^{\#,t}_{s} + \sum_{k,\hat{s}} x^{t-k,k}_{\hat{s},s} - \sum_{k,s'} x^{t,k}_{s,s'} + y^{-,t}_{s} - y^{+,t}_{s} = d^{\#,t+1}_{s} \quad ,\forall t, s \tag{15}$$

$$x^{t,k}_{s,s'} \leq d^{\#,t}_{s} \cdot \frac{F^{t,k}_{s,s'}}{\sum_{k,\hat{s}} F^{t,k}_{s,\hat{s}}}, \quad \forall t, k, s, s' \tag{16}$$

$$y^{+,t}_{s} + y^{-,t}_{s} \leq C^{*}_{v} \cdot Z^{t}_{s} \quad ,\forall t, s \tag{17}$$

$$\sum_{s \in \tilde{s} | z^{t}_{\tilde{s},v} = 1} [y^{+,t}_{s} - y^{-,t}_{s}] = d^{*,t+1}_{v} - d^{*,t}_{v} \quad ,\forall t, \tilde{s} \tag{18}$$

$$0 \leq x^{t,k}_{s,s'} \leq F^{t,k}_{s,s'}, 0 \leq y^{+,t}_{s}, y^{-,t}_{s} \leq C^{*}_{v}, 0 \leq d^{\#,t}_{s} \leq C^{\#}_{s} \tag{19}$$

Table 6: GETSTATIONREDEPLOY$(v, \mathbf{Z}, \mathbf{d}^{*}_{v})$

In the third step, 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. Let $\tilde{z}$ be the routing strategy obtained by solving the abstract problem, where in $\tilde{z}^{t}_{\tilde{s},v} = \sum_{\tilde{s}'} \tilde{z}^{t}_{\tilde{s},\tilde{s}',v} = 1$ entails carrier v is present in abstract state $\tilde{s}$ at time step t. Also, if s is an original station and $\tilde{s}$ is an abstract station, then let $Z^{t}_{s} = 1$, if $s \in \tilde{s}$ and $\sum_{v} \tilde{z}^{t}_{\tilde{s},v} = 1$.

$$\min_{\mathbf{z}} \sum_{t,s,s'} P_{s,s'} \cdot z^{t}_{s,s'} \tag{20}$$

$$\textbf{s.t. } \hat{d}^{*,t} + \sum_{s} (Y^{+}_{s} - Y^{-}_{s}) \sum_{s'} z^{t}_{s,s'} = \hat{d}^{*,t+1}, \forall t \in \hat{T} \tag{21}$$

$$\sum_{t,s'} z^{t}_{s,s'} = 1 \quad ,\forall s \in \tilde{s} | (Y^{+}_{s} + Y^{-}_{s}) > 0 \tag{22}$$

$$\sum_{s'} z^{t}_{s,s'} - \sum_{\hat{s}} z^{t-1}_{\hat{s},s} = \sigma^{t}(s) \quad ,\forall t \in \hat{T}, s \in \tilde{s} \tag{23}$$

Table 7: GETINTRAROUTING$(\tilde{s}, \mathbf{Y})$

The optimization problem of Table 6 employs the constants, $\mathbf{Z}$ to obtain a base station level redeployment strategy, $\mathbf{y}$. One of the key differentiating constraints that has not been used earlier is constraints (18). 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.

Given the base station level redeployment strategy, $\mathbf{Y}$, we now compute the best route within the stations of an abstract station, $\tilde{s}$, while visiting each base station (where a redeployment is required) once and satisfying the redeployment numbers from each station, $\mathbf{Y}$. This problem can be solved locally for each abstract station, $\tilde{s}$, where carrier v is redeploying at time step t. If we have $|T|$ time-step and $|\mathcal{V}|$ carriers, we can solve $|T| \cdot |\mathcal{V}|$ subproblems separately. To figure out the initial location, we find a station within the abstract state 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: 7 provides the optimization formulation to solve each subproblem.

Experimental Results

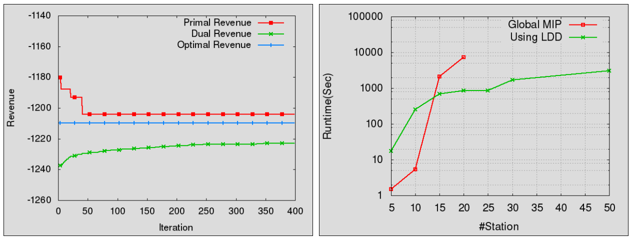
In this section, we evaluate our approaches with respect to run-time, revenue for company and lost demand on real world[5] and synthetic data sets. These data sets contain the following data: (1) Customer trip records that are indicative of successful bookings. We predict demand from these trip records. (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. (5) Carrier vehicles incur 1.5 USD for every 12 kms [6].

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 redeploying 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.

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

[6] Mileage results in Table 2 of (Fishman, Washington, and Haworth 2014). http://www.globalpetrolprices.com/diesel_prices/#USA, shows diesel prices.



(a) (b)

Figure 1: (a) Duality gap (b) Runtime: LDD vs Global MILP

We have three sets of results[7] on the synthetic data set. Firstly, we compare the runtime performance of LDD (SOLVELDD()) with the global MILP (SOLVEDRRP()) in Figure 1(b). 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. Except on small scale problems (ex: 5-10 stations), LDD outperforms global MILP with respect to runtime. More 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 within an hour.

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: 1(a) 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: 8 shows the effect of abstraction on the generated revenue and execution time based on five random instances of customer demand. Although, there is only a reduction of 0.2% on average from optimal, it gives a significant computational gain.

	With Abstraction		Without Abstraction	
Instance	**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 8: Effect of Abstraction

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).

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:9 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 during the 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 %

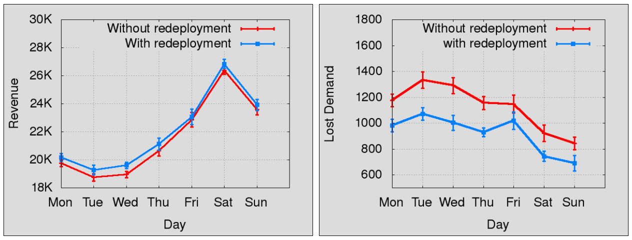
Table 9: Revenue and lost demand comparison

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:2 shows the mean and deviation of the revenue and lost call for each of the weekdays. Even considering the variance, Figure: 2(a) shows that the revenue generated by following our redeployment strategy is still better (albeit by a small amount) than current practice. More importantly, Figure: 2(b) demonstrates that we are able to significantly reduce the lost demand on all the cases.

We have done the same set of experiments with real-world data set of Hubway. Our approach is able to gain an excess 5% in revenue on average while the lost demand is reduced by a minimum of 40 %.

In summary, we have shown on multiple real and synthetic data sets, 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.

[7] All the linear optimization models were solved using CPLEX incorporated within python code on a 3.2 GHz Intel Core i5 machine with 4GB DDR3 RAM



(a) (b)

Figure 2: Sensitivity analysis: (a) Revenue comparison (b) Lost demand comparison

References

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.; Aravamudhan, 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.

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

Raidl, G. R.; Hu, B.; Rainer-Harbach, M.; and Papazek, P. 2013. Balancing bicycle sharing systems: Improving a vns by efficiently determining optimal loading operations. 130–143.

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 Hoeve, W.-J. 2013. Inventory rebalancing and vehicle routing in bike sharing systems.

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. *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*.