Dynamic Redeployment to Reduce Lost Demand in Vehicle Sharing Systems

Supriyo Ghosh[a], Pradeep Varakantham[a], Yossiri Adulyasak[b], Patrick Jaillet[c]

[a] *School of Information Systems, Singapore Management University, Singapore 178902*
[b] *Singapore-MIT Alliance for Research and Technology (SMART), Massachusetts Institute of Technology*
[c] *Department of Electrical Engineering and Computer Science, Massachussets Institute of Technology*

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.

Keywords: Assignment, Artificial Intelligence, Large-scale Optimization, OR in Bike-Sharing, Lagrangian Dual Decomposition

1. 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. STS have the ability to provide healthier living and greener environments while delivering fast movement for customers. 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. While we focus on bike-sharing systems in this paper, our model, methodology and evaluation generalize to other vehicle-sharing systems.

Bike sharing systems are widely adopted with 747 active systems and a fleet of over 772,000 bicycles and 235 systems are in planning or under construction [1], in major cities around the world. Figure 1 provides a quick view of the different bike sharing systems around the world.

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 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 the configuration at the beginning of the day or to some other pre-determined configuration.

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 2 provides the number of instances when stations become empty or full

Email addresses: supriyog.2013@phdis.smu.edu.sg (Supriyo Ghosh), pradeepv@smu.edu.sg (Pradeep Varakantham), yossiri@smart.mit.edu (Yossiri Adulyasak), jaillet@mit.edu (Patrick Jaillet)

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

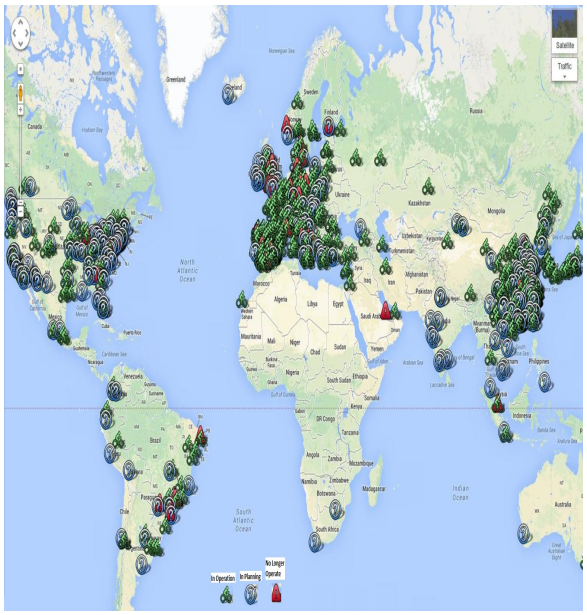

Figure 1: Visualization of the BSS worldwide

during the day (aggregated over 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 in 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). However, because carriers incur a cost in executing a given redeployment strategy, we have to consider the trade-off between our original problem of minimizing lost demand (alternatively maximizing revenue) and the consequent problem of minimizing cost incurred by carriers. 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 et al. 2010) and dynamic redeployment of emergency vehicles (Yue et al. 2012, Saisubramanian et al. 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. We have referred to these papers in the related work section. The key distinction from existing research is that we consider the dynamic redeployment of bikes in conjunction with the routing problem for carriers. Furthermore, we also provide novel approaches that employ aggregation of base stations and decomposability in DRRP optimization to solve demand loss problem for real world bike-sharing systems.

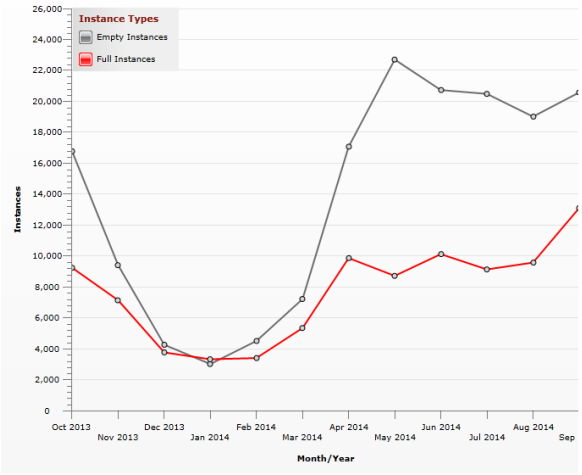

Figure 2: Number of empty/full instances of stations in CapitalBikeShare Company

DRRP is NP-Hard and therefore, we focus on developing principled approximation methods. To that end, here are our key contributions:
(1) A mixed integer and linear optimization formulation to maximize profit for the vehicle sharing company by trading off between two conflicting goals of:

- computing the optimal re-deployment strategy (i.e., how many vehicles have to be picked or dropped from each base station and when) for shared vehicles; **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 method to decompose the overall optimization formulation into two components – one which computes

re-deployment strategy for vehicles and one which computes routing policies for carriers.
(3) An abstraction mechanism that clusters the nearby base stations to reduce the size of decision problem and further 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 vehicle-sharing systems.

2. Motivation: Reducing Lost Demand in Bike Sharing Systems

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 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 flow of bikes between different stations, the goal is to redeploy the bikes by routing the carrier vehicles so as to maximize the overall profit[2] of the bike sharing company. As indicated earlier, we refer to this as the Dynamic Redeployment and Routing Problem (DRRP).

Example 2.1. *In Figure 3, we provide an example to better explain a DRRP. For the ease of explanation, we take 3 stations and movement of bikes between the stations is shown over 3 time-steps. An oval represents a station at a time step. The leftmost oval at the top is station 1 at time step 1, the oval in the middle column at the top is station 1 at time step 2 and so on. The number inside the oval represents the number of bikes present in the station at that time step. The numbers in the square box on top of each oval represent the actual demand in that station at that time-step. Numbers on each arc show the actual flow of bikes on that edge. Blue ellipse on the top of an oval represents the lost demand at that station due to the unavailability of bikes. For this specific example the total loss in demands is four as shown in Figure 3a. However, if we redeploy the idle bikes efficiently with the help of a carrier (route shown by the dotted line in the Figure 3b), then there is no lost demand.*

DRRP is a generalization of Static Bicycle Repositioning Problem [*SBRP*] (Schuijbroek et al. 2013) (redeployment at the end of day), which in turn can be reduced from the well known NP-Complete problem of 3-set partitioning (3PP). Due to this relation, it can be easily shown that DRRP is at least NP-Complete.

3. Optimization Model For Solving DRRP

We employ a data driven approach[3] 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 $\mathbf{F}$. 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

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

[3] Details of the data driven approach are provided in the discussion sub-section and experimental results section.

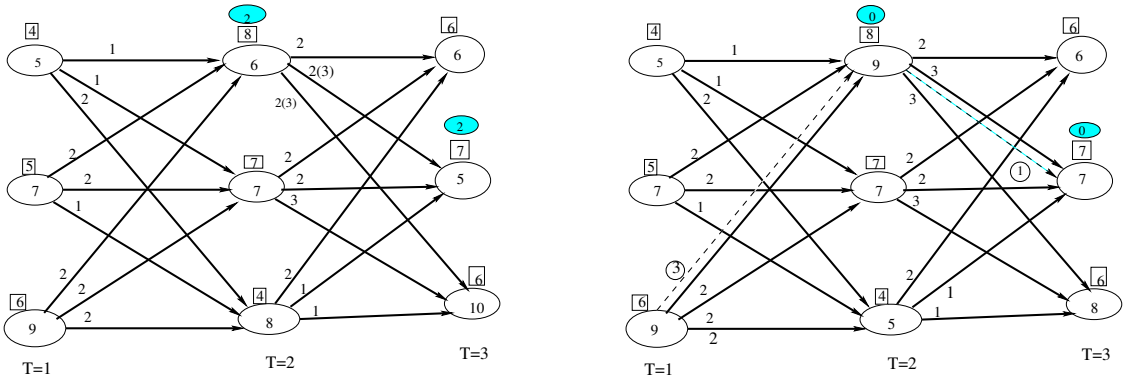

Figure 3: Synthetic model explaining the need of dynamic redeployment: (a) Without redeployment (Lost Demand = 4) (b) With redeployment(Dotted line represent the redeployment policy, Lost Demand = 0)

Category	Variable	Definition
Decision	$y^{+,t}_{s,v}$	Number of bikes picked from s by carrier v at time t
	$y^{-,t}_{s,v}$	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)
Interme-diate	$x^{t,k}_{s,s'}$	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**

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., $F^{t,k}_{s,s'}$. This is in the case where number of bikes at station s is lower than the demand. To represent this, we introduce a proxy variable, $x^{t,k}_{s,s}$ for $F^{t,k}_{s,s'}$ that is set based on 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, $\mathbf{F}$.

To represent the tradeoff 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. This objective is represented in Equation 1 of the MILP in SOLVEDRRP(). We have the following flow preservation, movement and capacity constraints for bikes, stations and carriers:

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

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

$$x^{t,k}_{s,s'} \le d^{\#,t}_{s}\cdot\frac{F^{t,k}_{s,s'}}{\sum_{k,\hat{s}} F^{t,k}_{s,\hat{s}}}, \qquad \forall t,k,s,s' \quad (3)$$

$$d^{*,t}_{v} + \sum_{s\in S}[(y^{+,t}_{s,v} - y^{-,t}_{s,v})] = d^{*,t+1}_{v}, \qquad \forall t,v \quad (4)$$

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

$$\sum_{j\in\mathcal{S},v\in\mathcal{V}} z^{t}_{s,j,v} \le 1, \qquad \forall t,s \quad (6)$$

$$y^{+,t}_{s,v} + y^{-,t}_{s,v} \le C^{*}_{v}\cdot\sum_{i} z^{t}_{s,i,v}, \qquad \forall t,s,v \quad (7)$$

$$0 \le x^{t,k}_{s,s'} \le F^{t,k}_{s,s'}, 0 \le d^{\#,t}_{s} \le C^{\#}_{s}, 0 \le y^{+,t}_{s,v}, y^{-,t}_{s,v} \le C^{*}_{v} \quad (8)$$

$$0 \le d^{*,t}_{v} \le C^{*}_{v}, z^{t}_{i,j,v} \in \{0,1\} \quad (9)$$

Table 2: SOLVEDRRP()

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 at station, s at time, $t+1$ (i.e., $d^{\#,t+1}_{s}$) to the sum of bikes at station, s at time, t (i.e., $d^{\#,t}_{s}$) and net number of bikes arriving at station s after time step t (i.e., $\sum_{k,\hat{s}} x^{t-k,k}_{\hat{s},s} - \sum_{k,s'} x^{t,k}_{s,s'}$), minus the net number of bikes picked up from station s by all

carriers v (i.e., $\sum_v \left[y^{-,t}_{s,v} - y^{+,t}_{s,v}\right]$).

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 (i.e. $d^{\#,t}_s$) and the transition probability that a bike will move from s to s' (according to customer demand (i.e. $F^{t,k}_{s,s'} / \sum_{k,\hat{s}} F^{t,k}_{s,\hat{s}}$)).
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 a carrier, v at time step $t+1$ (i.e., $d^{*,t+1}_v$) to the sum of bikes in carrier, v at time step t (i.e., $d^{*,t}_v$) and the net number bikes added at time step t to carrier v (i.e., $\sum_{s\in\mathcal{S}} \left[y^{+,t}_{s,v} - y^{-,t}_{s,v}\right]$).
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 (i.e., $\sum_{k\in\mathcal{S}} z^t_{s,k,v}$) , s for a carrier v at time t is equivalent to flow into the station s at time $t-1$ (i.e., $\sum_{k\in\mathcal{S}} z^t_{k,s,v}$)[4]. For $t = 0$, since p^0_v is given as input, this constraints will ensure carrier flow moves appropriately out of the initial locations.
5. *More than one carrier cannot be in one station at any time step*: Constraints (6) ensure this by restricting the maximum carrier flow in a station (i.e., $\sum_{k\in\mathcal{S}} z^t_{s,k,v}$) as one.
6. *Carrier can only pick up or drop off bikes from a station if they are at that station*: This is enforced using Constraints (7), where in, for each carrier v, the number of vehicles picked up or dropped off at a time is bounded by whether the station is visited at that time step. Also, the maximum number of bikes that can be redeployed at that time step is C^*_v.
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 vehicle 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 are always less than the capacity of the vehicle (i.e., C^*_v).

[4]Note that this constraint does not preclude a carrier from staying in the same station

$$\min_{\mathbf{y}^+,\mathbf{y}^-,\mathbf{z}} \; -\sum_{t,k,s,s'} R^{t,k}_{s,s'} \cdot x^{t,k}_{s,s'} + \sum_{\hat{t},v,s,s'} P_{s,s'} \cdot z^{\hat{t}}_{s,s',v} \tag{10}$$

$$\textbf{s.t. } d^{\#,t}_s + \sum_{k,\hat{s}} x^{t-k,k}_{\hat{s},s} - \sum_{k,s'} x^{t,k}_{s,s'} + \sum_{\hat{t}=m.(t)}^{m.(t+1)} \sum_v (y^{-,\hat{t}}_{s,v} - y^{+,\hat{t}}_{s,v}) = d^{\#,t+1}_s, \;\; \forall t, s \tag{11}$$

$$x^{t,k}_{s,s'} \le d^{\#,t}_s \cdot \frac{F^{t,k}_{s,s'}}{\sum_{k,\hat{s}} F^{t,k}_{s,\hat{s}}}, \qquad \forall t,k,s,s' \tag{12}$$

$$d^{*,\hat{t}}_v + \sum_{s\in S}[(y^{+,\hat{t}}_{s,v} - y^{-,\hat{t}}_{s,v})] = d^{*,\hat{t}+1}_v, \quad \forall \hat{t}, v \tag{13}$$

$$\sum_{k\in\mathcal{S}} z^{\hat{t}}_{s,k,v} - \sum_{k\in\mathcal{S}} z^{\hat{t}-1}_{k,s,v} = \sigma^{\hat{t}}_v(s), \qquad \forall \hat{t}, s, v \tag{14}$$

$$\sum_{j\in\mathcal{S}, v\in\mathcal{V}} z^{\hat{t}}_{s,j,v} \le 1, \qquad \forall \hat{t}, s \tag{15}$$

$$y^{+,\hat{t}}_{s,v} + y^{-,\hat{t}}_{s,v} \le C^*_v \cdot \sum_i z^{\hat{t}}_{s,i,v}, \qquad \forall \hat{t}, s, v \tag{16}$$

$$0 \le x^{t,k}_{s,s'} \le F^{t,k}_{s,s'}, 0 \le d^{\#,t}_s \le C^{\#}_s, 0 \le y^{+,\hat{t}}_{s,v}, y^{-,\hat{t}}_{s,v} \le C^*_v \tag{17}$$

$$0 \le d^{*,\hat{t}}_v \le C^*_v, z^{\hat{t}}_{i,j,v} \in \{0,1\} \tag{18}$$

Table 3: SOLVEDRRPDIFFTIMESCALES()

3.1. Discussion

While the formulation presented in Table 2 captures most aspects of bike flow dynamics in real world, there are few assumptions and boundary conditions:

- For ease of understanding and providing the intuitions, the MILP in Table 2 assumes that the time scale for movement of bikes and trucks is the same. That is to say, we assume that each carrier will redeploy bikes from at most one station in each time step. In practice, a carrier can redeploy bikes from multiple stations in each time-step. Therefore, we have a general formulation that has bikes and carriers operating on different time scales. Except for the two different time scales where a carrier is assumed to travel m stations at each time step (i.e., $t = m \cdot \hat{t}$), Table 3 provides the new MILP that is exactly similar in structure to the MILP in Table 2. Therefore, the enhancements provided with respect to decomposition and abstraction are applicable in similar ways.
- Since we maximize revenue, which is a multiple of **x**, we can identify boundary cases where bikes

are not issued at certain time steps even though demand is present. Such cases can arise to save bikes for a later time step when it is possible to get higher revenue. Since, we have a data driven approach where we evaluate on a test data set (that is different from training data set), accounting for real dynamics in training data set is not always necessary, more so, when accounting for exact dynamics increases the complexity of strategy computation significantly. This is demonstrated in our experimental results, where even with approximate dynamics, we are able to provide significant improvements over current practice.

To capture real dynamics we would have to introduce new set of constraints (19) that ensure total outflow of bikes from a station s at time t should be equal to the minimum of total arrival demand and the number of bikes present at source station. But constraints (19) are quadratic in nature and due to this our MILP becomes a higher order conic program.

$$\sum_{k,s'} x_{s,s'}^{t,k} = \min(d_s^{\#,t}, \sum_{k,s'} F_{s,s'}^{t,k}) \quad \forall t, s \qquad (19)$$

Apart from being quadratic, as mentioned in Shu et al. (2013), constraints (19) can only be the sufficient condition to handle real dynamic of BSS, if stations have unlimited bike docking capacity. Because of these difficulties, we focus on representing bike flow dynamics approximately.

- Given the non-trivial nature of problems even with expected demand values, considering variance in demand while computing routing and redeployment strategies is left for future work. It should however be noted that we evaluated our strategies computed from expected demand values on different demand scenarios and we show a significant improvement in performance across key performance indicators.

4. 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:*

- $\boldsymbol{y}^+$ *and* $\boldsymbol{y}^-$ *variables capture the solution for the redeployment problem.*
- $\boldsymbol{z}$ *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 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 the following contributions relevant to the specific problem:

1. **Identifying the right constraints to be dualized**[5]: If the right constraints are not dualized, then the underlying Lagrangian based optimization may not be decomposable or it may take significantly more time than the centralized MILP to find the desired solution.
2. **Extraction of primal solution from infeasible dual solution**[6]: In most mixed integer programs, solution generated from solving the dual decomposed slaves can be infeasible and hence the overall approach can potentially lead to slower convergence and bad solutions.

As demonstrated in the rest of the section, we are able to get the dual solution correctly and are also able to extract a primal solution from the dual solution (irrespective of its feasibility in the primal space).

In order to get 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) contain a dependency between routing and redeployment problems. Thus, we dualize constraints (7) using the price variables, $\alpha_{s,t,v}$ and obtain

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

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

Algorithm 1: SolveLDD($drrp$)

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

the Lagrangian as follows:

$$\mathcal{L}(\alpha) = \min_{\mathbf{z},\mathbf{y}^+,\mathbf{y}^-} \Big[- \sum_{t,k,s,s'} R^{t,k}_{s,s'} \cdot x^{t,k}_{s,s'} + \sum_{t,v,s,s'} P_{s,s'} \cdot z^t_{s,s',v} + \sum_{s,t,v} \alpha_{s,t,v} \cdot (y^{+,t}_{s,v} + y^{-,t}_{s,v} - C^*_v \cdot \sum_i z^t_{s,i,v}) \Big] \quad (20)$$

$$= \min_{\mathbf{y}^+,\mathbf{y}^-} \Big[- \sum_{t,k,s,s'} R^{t,k}_{s,s'} \cdot x^{t,k}_{s,s'} + \sum_{s,t,v} \alpha_{s,t,v} \cdot \big(y^{+,t}_{s,v} + y^{-,t}_{s,v}\big) \Big] + \min_{\mathbf{z}} \Big[\sum_{t,v,s,s'} P_{s,s'} \cdot z^t_{s,s',v} - \sum_{s,t,v} \alpha_{s,t,v} \cdot C^*_v \cdot \sum_i z^t_{s,i,v} \Big] \quad (21)$$

In Equation 21, 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 for DRRP by dualizing constraints 7. More specifically, the **slave optimizations** corresponding to the redeployment and routing problems are given in Table 4 and Table 5 respectively.

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

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

Table 4: SOLVEREDEPLOY()

From Equation 21, given an α, the dual value corresponding to the original problem is obtained by adding up the solution values from the two slaves. It should be noted that the decomposition is only for $\mathcal{L}(\alpha)$ and to obtain the final solution for the original optimization problem provided in Table 2, we have to solve the following optimization problem at the **master** in order to

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

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

Table 5: SOLVEROUTING()

reduce violation of the dualized constraint:

$$\max_{\alpha} \mathcal{L}(\alpha) \quad (22)$$

This **master** optimization problem is solved iteratively using sub-gradient descent on price variables, α:

$$\alpha^{k+1}_{s,t,v} = \big[\alpha^k_{s,t,v} + \gamma \cdot (y^{+,t}_{s,v} + y^{-,t}_{s,v} - C^*_v \cdot \sum_i z^t_{s,i,v})\big]_+ \quad (23)$$

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 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 the step size. 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 [7], 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) 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 $Z^t_{s,v} = \sum_{s'} z^t_{s,s',v}$. We extract the primal solution by solving the optimization formulation provided in Table 6. Essentially we solve the redeployment slave with an additional set of constraints (24), which ensure that redeployment in a station is possible if a carrier vehicle is present there. More specifically, constraints (24)

[7] The gap between dual and primal solution which is known as duality gap, is the measurement of solution quality derived from LDD. We reach an optimal solution if the duality gap becomes zero.

are equivalent to complicating constraints (7) where we inject the solution of routing slave (**z**) as input. Thus ExtractPrimal() satisfy all the constraints of Table: 2 and is a feasible solution to it. Finally, we subtract the routing cost from the objective to get the primal value.

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

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

$$y^{+,t}_{s,v} + y^{-,t}_{s,v} \leq C^*_v \cdot Z^t_{s,v} \quad ,\forall t, s, v \qquad (24)$$

Table 6: EXTRACTPRIMAL()

Proposition 1. *(Fisher 1985) : Error in solution quality obtained by Lagrangian dual decomposition method in Algorithm 1 is bounded by* $p - (o1 + o2)$.

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

5. 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 significantly higher. In order to provide scalable solutions to such problems, we propose a heuristic approach based on creating abstract stations, each of which represents a set of original base stations. We first solve the abstract problem and derive the solution to the original problem from the solution to the abstract problem. 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 —-

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 in the abstract DRRP remains the same as in the original DRRP. In practice, revenue, $R^{t,k}_{s,s'}$ 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[8] (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: $d^{\#,0}_{\tilde{s}} = \sum_{s\in\tilde{s}} d^{\#,0}_s$. 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 : $\sigma^0_{v,\tilde{s}} = 1$, if $\exists s \in \tilde{s}$ and $\sigma^0_{v,s} = 1$. 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: $F^{t,k}_{\tilde{s},\tilde{s}'} = \sum_{\{s\in\tilde{s},s'\in\tilde{s}'\}} F^{t,k}_{s,s'}$. 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.
- Routing 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 —-

In the second step, we use LDD from Section 4 to solve the abstract DRRP. As the abstract stations contain multiple base stations, we have to modify the constraints (6)

[8]The grouping of base stations can be done in various way and the results may vary for different problem instances. Clustering of base stations according to geographical proximity is one option and in the experimental results we have demonstrated that it provides a reasonable improvement over current practice. Other options are still feasible and our algorithm remains the same.

to allow multiple carriers in an abstract station. Table:7 provides the modified version of the routing slave to solve the abstracted DRRP. The modified set of Constraints (25) ensure that at any time step maximum $|\tilde{s}|$ number of carriers can visit in abstract station $\tilde{s}$. However, the redeployment slave and master function remain same. There are two key outputs from LDD: (a) Redeployment strategy, $\tilde{y}$ for moving bikes between abstract stations; and (b) Routing strategy, $\tilde{z}$ for moving between abstract stations, $\tilde{s}$ at different time steps.

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

s.t. *Constraints 5 & 9 hold*

$$\sum_{j\in\tilde{S},v\in\mathcal{V}} \tilde{z}^t_{\tilde{s},j,v} \leq |\tilde{s}|, \qquad \forall t,\tilde{s} \qquad (25)$$

Table 7: SOLVEABSTRACTEDROUTING()

Derive Strategies for Original DRRP —-
In the third step, we retrieve the solution of DRRP from abstract station level solution. $\tilde{z}$ is 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. Table 8 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 Table2 with an additional set of constraints (26) 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 $z^t_{s,s',v}$ can only be 1 if $s \in \tilde{s}$, $s' \in \tilde{s}'$ and $\tilde{z}^t_{\tilde{s},\tilde{s}',v} = 1$. In the MILP of RetrieveDRRP(), we explicitly set the decision variables $z^t_{s,s',v}$ to 0 if $s \in \tilde{s}$, $s' \in \tilde{s}'$ and $\tilde{z}^t_{\tilde{s},\tilde{s}',v} = 0$. Thus RetrieveDRRP() becomes an easier optimization problem than SolveDRRP().

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

s.t. *Constraints 2- 9 hold and*

$$\sum_{s\in\tilde{s},s'\in\tilde{s}'} z^t_{s,s',v} = \tilde{z}^t_{\tilde{s},\tilde{s}',v} \quad \forall \tilde{s},\tilde{s}',t,v \qquad (26)$$

Table 8: RETRIEVEDRRP()

5.1. 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. This would be the optimization problem provided in Table 3. Abstract version of this problem can be solved directly using the LDD from Section 4. However, the mechanism to retrieve the DRRP solution from abstract station level solution changes and is outlined here.

5.1.1. Base station level redeployment solution

We first compute redeployment strategy, **y** 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 9 employs the constants, **Z** to obtain a base station level redeployment strategy, **y**. 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 $\sum_v \tilde{z}^t_{\tilde{s},v} = 1$. Thus we explicitly specify that redeployment in a base station s is possible only if it belongs to an abstract station $\tilde{s}$, where a carrier is present in the abstract solution. Our objective function specified in (27) is to maximize the overall revenue of the agency that implies maximum number of customers are served. Intuitively we have the following set of constraints:

1. *Flow preservation of bikes in and out of stations:* Constraints (28) enforce this by ensuring equivalence of the number of bikes at station, s at time, $t+1$ (i.e., $d^{\#,t+1}_s$) to the sum of bikes at station, s at time, t (i.e., $d^{\#,t}_s$) and net number of bikes arriving at station s after time step t (i.e., $\sum_{k,\hat{s}} x^{t-k,k}_{\hat{s},s} - \sum_{k,s'} x^{t,k}_{s,s'}$), minus the net number of bikes picked up from station s (i.e., $\left[y^{-,t}_s - y^{+,t}_s\right]$).
2. *Flow of bikes between any two stations follows the transition dynamics observed in the data:* Constraints (29) 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. *Redeployment from/to a station is possible if a carrier is present in that stations:* Constraints (30) ensure that redeployment from/to a station s is possible only if it belongs to an abstract station where a carrier is present in that time-step (i.e., $Z^t_s = 1$).
4. *Total bikes redeployed from the base stations of an abstract station is preserved:* Constraints (31) ensure that total bikes picked up or dropped off by

the carrier vehicle v, from all base stations in an abstract station $\tilde{s}$ (i.e., $\sum_{s\in\tilde{s}|z^t_{\tilde{s},v}=1}[y^{+,t}_s - y^{-,t}_s]$) is equal to the number of bikes picked up or dropped off by the carrier v at time t in the abstraction station level redeployment strategy (i.e., $d^{*,t+1}_v - d^{*,t}_v$).

5. *Station capacity is not violated during redeployment:* Constraints (32) limit the number of bikes present in a station s to be less than its capacity (i.e., $C^{\#}_s$).

$$\max_{\mathbf{y}^+,\mathbf{y}^-} \sum_{t,s,s,s'} R^{t,k}_{s,s'} \cdot x^{t,k}_{s,s'} \tag{27}$$

$$\textbf{s.t. } d^{\#,t}_s + \sum_{k,\hat{s}} x^{t-k,k}_{\hat{s},s} - \sum_{k,s'} x^{t,k}_{s,s'} + y^{-,t}_s - y^{+,t}_s = d^{\#,t+1}_s \quad ,\forall t,s \tag{28}$$

$$x^{t,k}_{s,s'} \le d^{\#,t}_s \cdot \frac{F^{t,k}_{s,s'}}{\sum_{k,\hat{s}} F^{t,k}_{s,\hat{s}}}, \qquad \forall t,k,s,s' \tag{29}$$

$$y^{+,t}_s + y^{-,t}_s \le C^*_v \cdot Z^t_s \quad ,\forall t,s \tag{30}$$

$$\sum_{s\in\tilde{s}|z^t_{\tilde{s},v}=1} [y^{+,t}_s - y^{-,t}_s] = d^{*,t+1}_v - d^{*,t}_v \, ,\forall t,\tilde{s} \tag{31}$$

$$0 \le x^{t,k}_{s,s'} \le F^{t,k}_{s,s'}, y^{+,t}_s, y^{-,t}_s \le C^*_v, d^{\#,t}_s \le C^{\#}_s \tag{32}$$

Table 9: GETSTATIONREDEPLOY($v, \mathbf{Z}, \mathbf{d}^*_v$)

5.1.2. Base station level routing solution

Though the exact redeployment policy is already known to us, it is important to find the exact routing policy of the carriers within each abstract station. Our goal is to find the best route within the abstract station that should obey the redeployment policy known to us and visit each station only once. We need to consider two types of route (a) The best route within the abstract station (i.e. intra-abstract station route) (b) The shortest route among the abstract stations (i.e. inter-abstract station route).

Given the base station level redeployment strategy, $\mathbf{Y}$ (= $\mathbf{y}$), we now compute the best route within the stations of an abstract station, $\tilde{s}$, while visiting each base station 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. Thus, if we have $|T|$ time-step and $|\mathcal{V}|$ carriers, we can solve $|T| \cdot |\mathcal{V}|$ subproblems separately to get the intra-abstract station routing.

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 are known at first time step, we can always obtain the starting location. Furthermore, this incremental method will also minimize the inter-abstract station routing.

Table 10 provides the optimization problem to solve each subproblem of the intra-abstract station routing. The objective (delineated in 33) is to minimize the routing cost of the carrier. The constraints are defined as follows:

1. *Flow of bikes in and out of a carrier is preserved:* Constraints (34) enforce this by ensuring the equivalence between number of bikes present in the carrier at time $t+1$ to the sum of bikes present in the carrier at time t and number of bikes picked up from a station s at that time step (i.e., $Y^+_s - Y^-_s | \sum_{s'} z^t_{s,s'} = 1$).
2. *Each station is visited only once:* Constraints (35) restrict that each base station where a redeployment is required (i.e. $Y^+_s/Y_s > 0$) is visited only once.
3. *Flow preservation of carrier in and out of a station:* Constraints (36) ensure that the flow in to a station s (i.e., $\sum_{s'} z^{t-1}_{s',s}$) and flow out from that station at time t (i.e., $\sum_{s'} z^t_{s,s'}$) are equal. σ^0 represent the initial location of the carrier so that the carrier moves appropriately from the initial location.
4. *Capacity of the carrier vehicle is not exceeded during redeployment:* Constraints (37) ensure that the number of bikes picked up or dropped off by the carrier in aggregate does not exceed the capacity of the carrier (i.e. C^*_v).

Example 5.1. *Figure 4 provides an example to illustrate the abstraction method where a carrier can cover all stations within an abstract station in one time step. Solid circles represent the base stations and big circles represent the abstract stations. We consider a problem with 13 base stations and grouped them into three abstract stations (with 5 stations in abstract station 1 and 4 stations each in abstract stations 2 and 3). Initial location of a carrier is indicated with a circle over the solid circle. Figure 4a depicts the optimal abstract station level routing solution (by solving the LDD based global MIP on the abstract DRRP) for the carrier. Figure 4b depicts the base station level routing*

$$\min_{\mathbf{z}} \sum_{t,s,s'} P_{s,s'} \cdot z^t_{s,s'} \quad (33)$$

$$\textbf{s.t. } \hat{d}^{*,t} + \sum_s (Y^+_s - Y^-_s) \cdot \sum_{s'} z^t_{s,s'} = \hat{d}^{*,t+1} \quad ,\forall t \in \hat{T} \quad (34)$$

$$\sum_{t,s'} z^t_{s,s'} = 1 \quad ,\forall s \in \tilde{s} | (Y^+_s + Y^-_s) > 0 \quad (35)$$

$$\sum_{s'} z^t_{s,s'} - \sum_{\hat{s}} z^{t-1}_{\hat{s},s} = \sigma^t(s) \quad ,\forall t \in \hat{T}, s \in \tilde{s} \quad (36)$$

$$0 \leq \hat{d}^{*,t} \leq C^*_v, z^t_{s,s'} \in \{0,1\} \quad (37)$$

Table 10: GETINTRAROUTING($\tilde{s}, \mathbf{Y}$)

policy within the abstract station 1 at initial time step. It also shows the route from the exit station of abstract station 1 to its nearest station in abstract station 2. By this incremental process we find the base station level routing policy for the carriers. Figure 4c, 4d depicts the base station level routing policy within abstract station 2 and 3 respectively.

6. Experimental Settings

In this section, we evaluate our approaches with respect to run-time, profit, revenue for company and lost demand on real world[9] and synthetic data sets . These data sets contain the following data: (1) Customer trip history records that provide information about successful bike booking. We predict the actual demand from this trip history. (2) Total 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 location of the stations. From the longitude and latitude information of stations, we calculate the relative distance between two stations. (4) Revenue model of the agency[10]. (5) Overestimation (to consider the worst case) of the cost of fuel for carriers[11].

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. (b) We take the same revenue and cost model discussed earlier from real datasets. We employ synthetic data sets to primarily illustrate the performance benefits provided by LDD over basic MILP based solution. This is because both LDD and MILP are unable to solve problems at the scale of the real world data sets.

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.

6.1. Simulating Flow of Bikes

In this section we describe the simulation model (that represents the flow of bikes in the real world) employed to determine how bikes flow between base stations. $F^{t,k}_{s,s'}$ denotes the outgoing demand from station s to s' at time t and $d^{\#,t}_s$ is the number of bikes present is station s at time t. The flow of bikes is determined based on the following two cases: (a) If total outflow from a base station (demand) is less than the number of bikes present in the base station, then all the customers are served. (b) If total outflow from a a station is higher than the number of bikes present in the base station, then actual flow (denoted as $x^{t,k}_{s,s'}$) depends on the relative ratio $\frac{F^{t,k}_{s,s'}}{\sum_{s',k} F^{t,k}_{s,s'}}$.

$$x^{t,k}_{s,s'} = \left\{ \begin{array}{ll} F^{t,k}_{s,s'} & \textbf{if } \sum_{k,s'} F^{t,k}_{s,s'} \leq d^{\#,t}_s \\ \frac{F^{t,k}_{s,s'}}{\sum_{k,\tilde{s}} F^{t,k}_{s,\tilde{s}}} \cdot d^{\#,t}_s & \textbf{Otherwise} \end{array} \right\}$$

We calculate the number of bikes present in a station at time $t+1$ (see eqn: 38)as the sum of left out bikes at time t and net incoming bikes in that time-step.

$$d^{\#,t+1}_s = d^{\#,t}_s + \Big[\sum_{k,\tilde{s}} x^{t-k,k}_{\tilde{s},s} - \sum_{k,s'} x^{t,k}_{s,s'} \Big] \quad (38)$$

So, in each time-step we determines the actual flow of bikes and then with that information we generate the distribution of bikes in the next time-step.

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

[10] 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 or rides.

[11] Mileage results are shown in Table 2 of Fishman et al. (2014) and http://www.globalpetrolprices.com/diesel_prices/#USA, shows that diesel price is 1.01 USD per liter, but we overestimate it as 1.5 USD to include the cost for extra resource (ex: petrol) consumption.

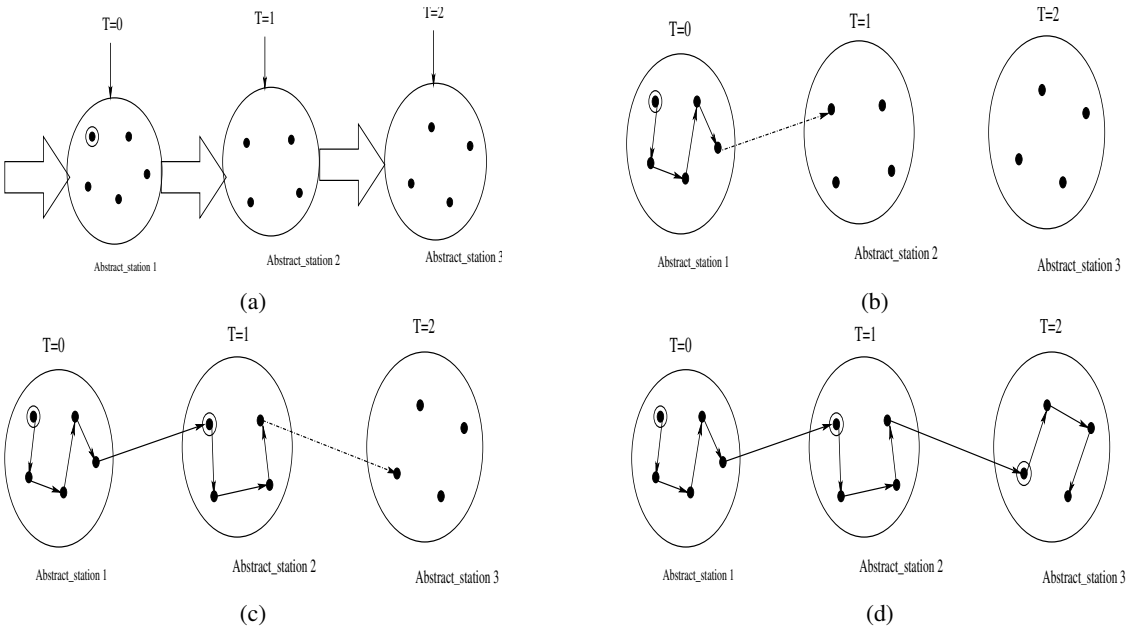

Figure 4: Routing solution in Abstracted DRRP: (a) Abstract station level routing solution (b) Routing solution within abstract station 1 (c) Routing solution within abstract station 2 (d) Routing solution within abstract station 3

6.2. Benchmark Approach: Current Practice

As indicated earlier, currently, the bike sharing systems redeploy bikes only at the end of the day. We compare our approaches against this current practice of not redeploying bikes during the day. Using the simulation model described in Section 6.1, we compute the flow of bikes when no redeployment is done and use that flow information to calculate the revenue and lost demand.

Since, we use the simulation approach of Section 6.1 to also evaluate the redeployment strategy computed by our decomposition and abstraction approaches, the comparison against current practice is fair.

7. Experimental Results

We have three sets of results[12] on the synthetic data set. Firstly, we compare the runtime performance of LDD (SOLVELDD()) with the global MILP (SOLVEDRRP()) in Figure 5b. 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 in 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 5a shows that the duality gap for a 20 station problem is only 1%. Figure 5c,5d depict the duality gap for real world data set of Hubway (with 95 base stations and grouped in 25 abstract stations) and CapitalBikeShare (with 305 base stations and grouped into 50 abstract stations) respectively. For both of these 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 11 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.

Capital-Bikeshare Results ——
The majority of our results are provided on the *Capital-BikeShare* dataset. The data set has 305 active stations and we consider 50 abstract stations (obtained through

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

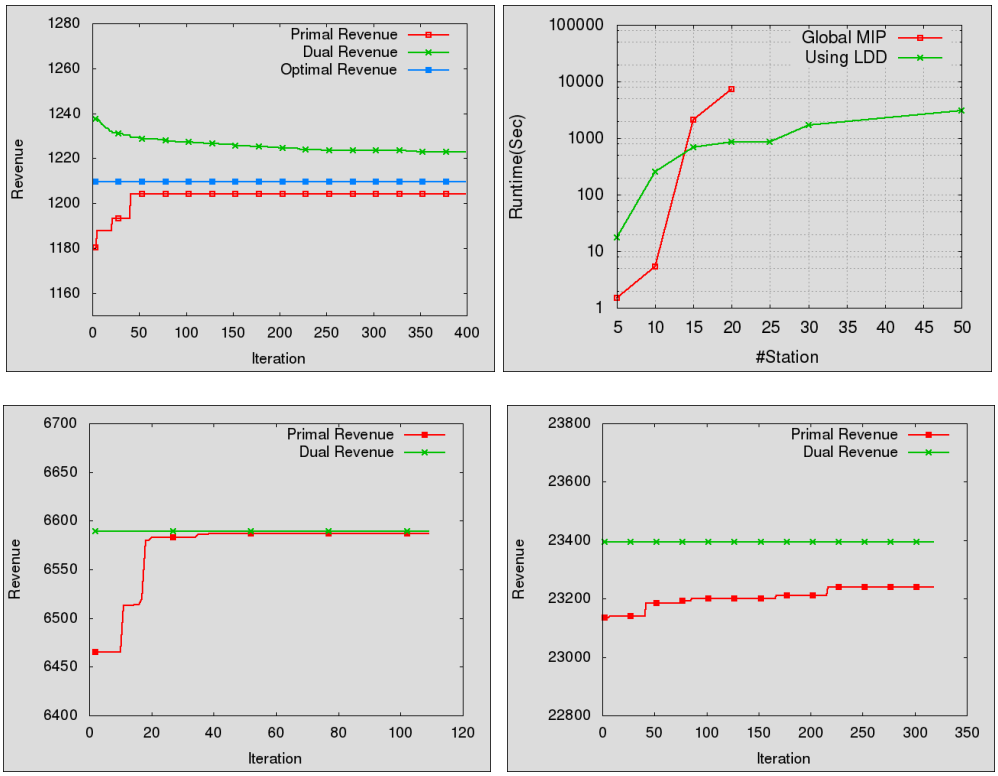

Figure 5: (a) Duality gap in synthetic dataset with 20 station (b) Runtime: LDD vs Global MILP (c) Duality gap in Hubway dataset (d) Duality gap in CapitalBikeshare dataset

	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 11: Effect of Abstraction

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 (from 5 AM to 12 AM) and also for one of the peak duration (from 5 AM to 12 PM). We consider the trip history data of four quarters of 2013 and for each quarter we have done the same set of experiments. Table12 shows the percentage gain in revenue and the percentage reduction in lost demand in comparison with current practice for the first quarter. 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 15%, a significant improvement over current practice. As expected, for all of those instances, the percentage revenue gain in the peak hours is much higher because most of the lost demands occurs in the peak hours.

Table13 shows the percentage gain in revenue and the percentage reduction in lost demand in comparison with current practice for the second quarter. Our approach is able to reduce the lost demand in all the cases by at least 10%, while the revenue is improved by an average of 3%.

Table14 shows the comparison with current practice

	Whole day (5am-12am)		Peak period (5am-12pm)	
	Revenue gain	**Lost demand reduction**	**Revenue gain**	**Lost demand reduction**
Mean	2.11 %	23.4 %	2.92 %	17.16 %
Mon	1.33 %	23.18 %	6.57 %	35.11 %
Tue	1.91 %	26.33 %	6.63 %	35.37 %
Wed	2.19 %	29.26 %	8.13 %	39.68 %
Thu	2.5 %	30.66 %	7.63 %	38.5 %
Fri	2.43 %	31.27 %	7.22 %	37.46 %
Sat	1.34 %	15.1 %	1.08 %	22.91 %
Sun	1.01 %	15.44 %	0.49 %	14.63 %

Table 12: Revenue and lost demand comparison (Dataset: Capital-Bikeshare, 1st quarter of 2013)

	Whole day (5am-12am)		Peak period (5am-12pm)	
	Revenue gain	**Lost demand reduction**	**Revenue gain**	**Lost demand reduction**
Mean	3.87 %	11.91 %	3.61 %	13.84 %
Mon	3.13 %	31.73 %	4.31 %	29.32 %
Tue	3.08 %	28.76 %	7.12 %	34.06 %
Wed	2.86 %	23.3 %	5.79 %	25.85 %
Thu	3.62 %	34.71 %	7.27 %	34.33 %
Fri	2.76 %	32.53 %	4.55 %	26.98 %
Sat	4.82 %	15.69 %	5.02 %	18.26 %
Sun	4.52 %	20.86 %	5.59 %	30.08 %

Table 13: Revenue and lost demand comparison (Dataset: Capital-Bikeshare, 2nd quarter of 2013)

	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 14: Revenue and lost demand comparison (Dataset: Capital-Bikeshare, 3rd quarter of 2013)

	Whole day (5am-12am)		Peak period (5am-12pm)	
	Revenue gain	**Lost demand reduction**	**Revenue gain**	**Lost demand reduction**
Mean	2.06 %	29.67 %	3.09 %	17.69 %
Mon	1.46 %	20.27 %	4.06 %	27.77 %
Tue	1.35 %	22.9 %	3.92 %	27.48 %
Wed	2.37 %	27.48 %	5.82 %	31.23 %
Thu	1.77 %	24.88 %	7.45 %	40.3 %
Fri	1.49 %	25.72 %	4.3 %	29.08 %
Sat	1.57 %	13.86 %	1.63 %	17.6 %
Sun	0.85 %	11.61 %	0.89 %	7.56 %

Table 15: Revenue and lost demand comparison (Dataset: Capital-Bikeshare, 4th quarter of 2013)

for the third quarter (busiest quarter in the year). For this quarter, our approach is able to reduce the lost demand in all the cases by at least 22%, while the revenue is improved by at least 3%.

Table15 shows the percentage gain in revenue and the percentage reduction in lost demand in comparison with current practice for the last quarter. For this data set, our approach reduces the lost demand by at least 10% over current practice. For all of those quarters, the percentage revenue gain in the peak hours is almost double because most of the lost demands occur in these time-period.

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 current practice. Figure 6 show the mean and deviation of the revenue and lost demand for the first quarter. Even considering the variance, Figure 6a depict that the revenue generated by following our redeployment strategy for each of the weekdays is still better (albeit by a small amount) than current practice. Figure 6b demonstrate that we are able to significantly reduce the lost demand on all the cases. It is interesting to note that the gain in weekdays (Mon-Fri) is significantly high because of a consistent pattern in customer demand. While in weekends, the demand may vary abruptly and the performance gain for all realization of demand is poor.

Figure 7 shows the mean and deviation of the revenue and lost call for the second quarter. Figure 7a depicts that the revenue generated by following our redeployment strategy for each of the weekdays is better than current practice. More interestingly, Figure 7b shows that we are able to significantly reduce the lost demand

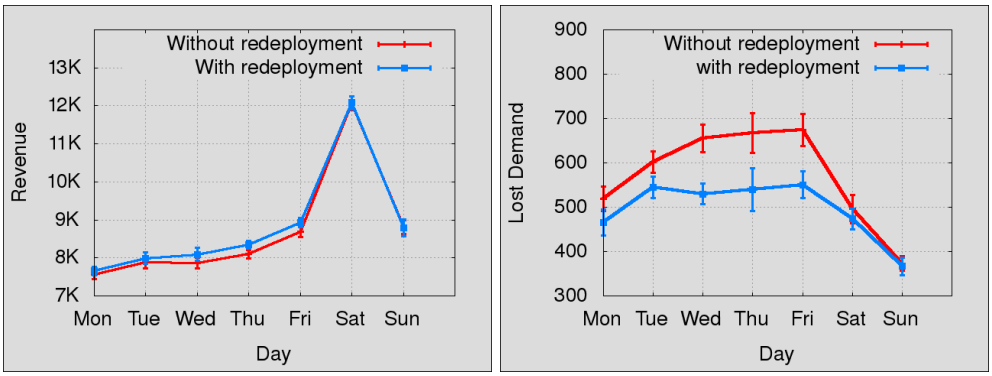

Figure 6: Sensitivity analysis [Dataset: CapitalBikeShare, 1st quarter of 2013]: (a) Revenue comparison (b) Lost demand comparison

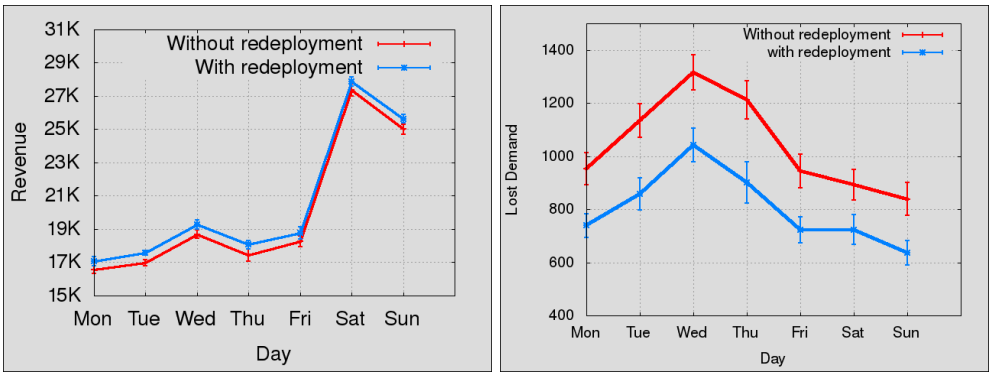

Figure 7: Sensitivity analysis [Dataset: CapitalBikeShare, 2nd quarter of 2013]: (a) Revenue comparison (b) Lost demand comparison

on all the instances.

Finally, we provide the mean and deviation of the revenue and lost call for the third and fourth quarter in Figure 8,9 respectively. Even considering the variance, Figure 8a,9a depict that the revenue generated by following our redeployment strategy for each of the weekdays is still better than current practice. Figure 8b,9b demonstrate that we are able to significantly reduce the lost demand on all the cases.

Hubway Results ——

Now we present the results on the real world dataset of *Hubway*. *Hubway* BSS comprises with 95 active stations and we group them into 25 abstract stations. We take two quarters (2nd and 3rd quarter of 2012) of trip history data into our consideration. Similar to the previous data set, we generate the overall mean demand as well as the demand for individual weekdays from historical data of trips.

We produce two sets of results with this data set. Table 16 provide the comparison results (on revenue and lost demand) for the third quarter. Our approach is able to gain an excess 5% in revenue on average while the lost demand is reduced by a minimum of 40 %.

	Without Redeployment		With Redeployment		Gain(%)	
	Revenue	**Lost demand**	**Revenue**	**Lost demand**	**Revenue**	**Lost demand**
Mean	5867	341	6342	92	8.10	73.02
Mon	6122	312	6363	179	3.94	42.63
Tue	5701	308	6039	121	5.93	60.71
Wed	6025	258	6293	107	4.45	58.53
Thu	6029	351	6385	159	5.90	54.70
Fri	5567	251	5916	57	6.27	77.29
Sat	6726	93	6874	28	2.20	69.89
Sun	6447	100	6650	26	3.15	74.00

Table 16: Revenue and lost demand comparison (Hubway, Dataset: 3-rd quarter of 2012)

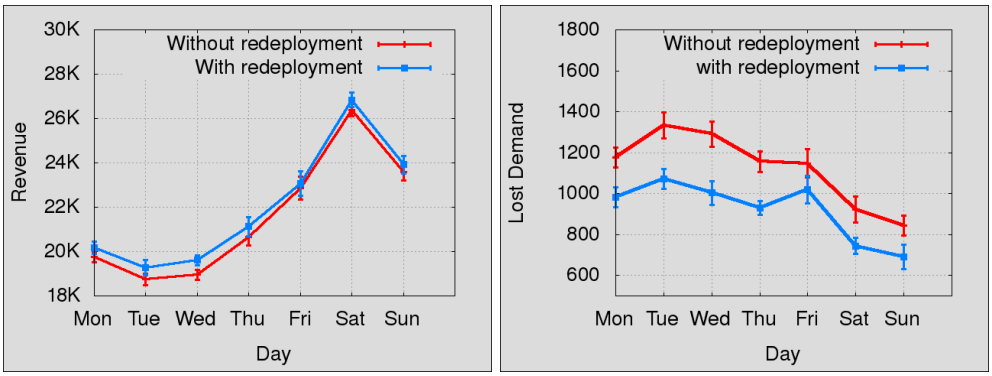

Figure 8: Sensitivity analysis [Dataset: CapitalBikeShare, 3rd quarter of 2013]: (a) Revenue comparison (b) Lost demand comparison

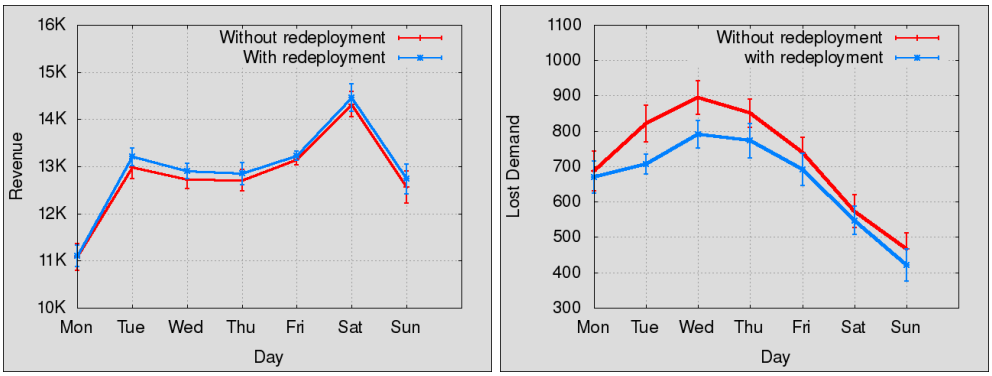

Figure 9: Sensitivity analysis [Dataset: CapitalBikeShare, 4th quarter of 2013]: (a) Revenue comparison (b) Lost demand comparison

In Table 17, we provide the percentage gain in revenue and the percentage reduction in lost demand in comparison with current practice for the second quarter of Hubway Dataset. In this scenario, our approach is able to gain an excess 3% in revenue on average while the lost demand is reduced by a minimum of 30 %.

Lastly, to visualize the effect of redeployment we draw the correlation between actual demand and served demand over the planning horizon. Since we aim to reduce the lost customer demand, it is better if most of the points are near to line of equality or identity line. Figure 10b 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: 10a).

	Without Redeployment		With Redeployment		Gain(%)	
	Revenue	**Lost demand**	**Revenue**	**Lost demand**	**Revenue**	**Lost demand**
Mean	4213	152	4317	106	2.46	30.26
Mon	4339	95	4488	30	3.43	68.42
Tue	4607	98	4751	33	3.12	66.32
Wed	3514	150	3743	40	6.51	73.33
Thu	4420	97	4504	61	1.90	37.11
Fri	4480	75	4549	42	1.54	44
Sat	4679	92	4792	43	2.41	53.26
Sun	4086	105	4232	41	3.57	60.95

Table 17: Revenue and lost demand comparison (Hubway, Dataset: 2-nd quarter of 2012)

8. 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

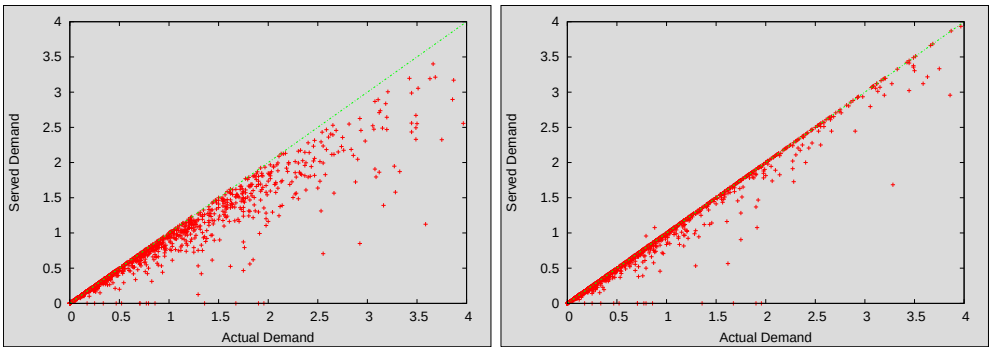

Figure 10: Correlation of demand and supply: (a) Without redeployment (b) With dynamic redeployment

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 et al. (2013) propose a scalable approximate solution for this problem by abstracting base stations into mega stations and solved it using a clustered vehicle routing (CluVRP) approach. Benchimol et al. (2011) present an approximate solution (inspired by the solution of *C-delivery TSP*) to solve the static re-balancing and routing problem by considering a single carrier vehicle. Raviv and Kolka (2013), Raviv et al. (2013) have provided scalable exact and approximate approaches to this routing problem by employing insights from inventory management. They introduce a set of MILP formulations that solve the static bicycle repositioning problem (SBRP). They penalize the objective function for unavailability of bikes or open docks and thus find an static redeployment model for the carriers to rearrange the inventory. Raidl et al. (2013) provide an approximate solution of SBRP by using variable neighbourhood 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. However, as described in Fricker and Gast (2012) and as demonstrated in our experimental results section, not performing redeployment during the day can result in a significant loss in demand. In contrast, our approaches differ from this thread of research as we consider dynamic redeployment 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. Kumar and Bierlaire (2012) develop a linear regression model to understand the correlation between stations location and customers location by analyzing the customers demographic and personal information. In addition, Martinez et al. (2012) provide an MILP formulation to optimally locate the stations, and solve it using branch and bound algorithm. Lin and Yang (2011), Lin et al. (2013) propose a decision support model to design and manage the public bicycle sharing (PBS) system. They provide a formal model to meet the service level requirement of PBS with the help of well known constraints from inventory or logistic management. Unfortunately, the flow in PBS depends on the stochastic demand or involuntary movement of customers and this static solution can significantly falter. 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 customer. 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. Chemla et al. (2013) solve the static re-balancing problem by considering a single carrier and provide a branch and cut algorithm that can solve a large scale problem with more than hundred stations. Contardo et al. (2012) propose a dynamic redeployment model 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. Though due to the complex structure of the problem there was a significant gap between their solution and its lower bound. 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 redeploy-

ment policy.

The third thread of research which is complementary to the work presented in this paper is on demand prediction and analysis. We have already highlighted the assumption that demand follows poisson distribution employed by Shu et al. (2013, 2010). Nair and Miller-Hooks (2011), Nair et al. (2013) provide 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) report the bike sharing system as a dual markovian waiting system to predict the actual demand. Froehlich et al. (2008), Lathia et al. (2012) predict the temporal user demand in terms of available bicycles or normalized available bicycle (NAB) in a station at a certain time step. Given its generality and can be employed over an entire horizon, we also employ the demand prediction model employed by Shu et al. (2013, 2010) and assume that demand follows a poisson distribution. However, the parameter, λ that governs the poisson distribution is learned from real data.

9. Conclusion

In this paper we address 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

C. Fricker, N. Gast, Incentives and redistribution in homogeneous bike-sharing systems with stations of finite capacity, EURO Journal on Transportation and Logistics (2012) 1–31.

J. D. Lees-Miller, J. C. Hammersley, R. E. Wilson, Theoretical maximum capacity as benchmark for empty vehicle redistribution in personal rapid transit, Transportation Research Record: Journal of the Transportation Research Board 2146 (1) (2010) 76–83.

Y. Yue, L. Marla, R. Krishnan, An Efficient Simulation-Based Approach to Ambulance Fleet Allocation and Dynamic Redeployment., in: AAAI, 2012.

S. Saisubramanian, P. Varakantham, L. H. Chuin, Risk based Optimization for Improving Emergency Medical systems., in: AAAI, 2015.

J. Schuijbroek, R. Hampshire, W.-J. van Hoeve, Inventory rebalancing and vehicle routing in bike sharing systems, Working Paper, Carnegie Mellon University .

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

M. L. Fisher, An applications oriented guide to Lagrangian relaxation, Interfaces 15 (2) (1985) 10–21.

G. J. Gordon, P. Varakantham, W. Yeoh, H. C. Lau, A. S. Aravamudhan, S.-F. Cheng, Lagrangian relaxation for large-scale multi-agent planning, in: Web Intelligence and Intelligent Agent Technology (WI-IAT), 2012 IEEE/WIC/ACM International Conferences on, vol. 2, IEEE, 494–501, 2012.

E. Fishman, S. Washington, N. L. Haworth, 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, 2014.

M. Benchimol, P. Benchimol, B. Chappert, A. De La Taille, F. Laroche, F. Meunier, L. Robinet, Balancing the stations of a self service bike hire system, RAIRO-Operations Research 45 (01) (2011) 37–61.

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

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

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

V. P. Kumar, M. Bierlaire, Optimizing locations for a vehicle sharing system, in: Swiss Transport Research Conference. http://www. strc. ch/conferences/2012/Kumar_Bierlaire. pdf, 2012.

L. M. Martinez, L. Caetano, T. Eiró, F. Cruz, An optimisation algorithm to establish the location of stations of a mixed fleet biking system: an application to the city of Lisbon, Procedia-Social and Behavioral Sciences 54 (2012) 513–524.

J.-R. Lin, T.-H. Yang, Strategic design of public bicycle sharing systems with service level constraints, Transportation research part E: logistics and transportation review 47 (2) (2011) 284–294.

J.-R. Lin, T.-H. Yang, Y.-C. Chang, A hub location inventory model for bicycle sharing system design: Formulation and solution, Computers & Industrial Engineering 65 (1) (2013) 77–86.

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

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

C. Contardo, C. Morency, L.-M. Rousseau, Balancing a dynamic public bike-sharing system, vol. 4, CIRRELT, 2012.

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

R. Nair, E. Miller-Hooks, R. C. Hampshire, A. Bušić, Large-Scale Vehicle Sharing Systems: Analysis of Vélib', International Journal of Sustainable Transportation 7 (1) (2013) 85–106.

F. Leurent, Modelling a vehicle-sharing station as a dual waiting system: stochastic framework and stationary analysis .

J. Froehlich, J. Neumann, N. Oliver, Measuring the pulse of the city through shared bicycle programs, Proc. of UrbanSense08 (2008) 16–20.

N. Lathia, S. Ahmed, L. Capra, Measuring the impact of opening the London shared bicycle scheme to casual users, Transportation research part C: emerging technologies 22 (2012) 88–102.