Journal of Artificial Intelligence Research 58 (2017) 387-430 Submitted 08/16; published 02/17

Dynamic Repositioning to Reduce Lost Demand in Bike Sharing Systems

Supriyo Ghosh SUPRIYOG.2013@PHDIS.SMU.EDU.SG
Pradeep Varakantham PRADEEPV@SMU.EDU.SG
School of Information Systems
Singapore Management University, Singapore 178902

Yossiri Adulyasak YOSSIRI.ADULYASAK@HEC.CA
Department of Logistics and Operations Management
HEC Montréal, 3000 chemin de la Côte-Sainte-Catherine
Montréal, H3T 2A7, Canada

Patrick Jaillet JAILLET@MIT.EDU
Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology, Cambridge
Massachusetts 02139, USA

Abstract

Bike Sharing Systems (BSSs) are widely adopted in major cities of the world due to concerns associated with extensive private vehicle usage, namely, increased carbon emissions, traffic congestion and usage of non-renewable resources. In a BSS, base stations are strategically placed throughout a city and each station is stocked with a pre-determined number of bikes at the beginning of the day. Customers hire the bikes from one station and return them at another station. Due to unpredictable movements of customers hiring bikes, there is either congestion (more than required) or starvation (fewer than required) of bikes at base stations. Existing data has shown that congestion/starvation is a common phenomenon that leads to a large number of unsatisfied customers resulting in a significant loss in customer demand. In order to tackle this problem, we propose an optimisation formulation to reposition bikes using vehicles while also considering the routes for vehicles and future expected demand. Furthermore, we contribute two approaches that rely on decomposability in the problem (bike repositioning and vehicle routing) and aggregation of base stations to reduce the computation time significantly. Finally, we demonstrate the utility of our approach by comparing against two benchmark approaches on two real-world data sets of bike sharing systems. These approaches are evaluated using a simulation where the movements of customers are generated from real-world data sets.

1. Introduction

Bike Sharing Systems (BSSs) offer attractive alternatives to private transportation particularly in alleviating concerns associated with increased carbon emissions, traffic congestion and usage of non-renewable resources. BSSs have the ability to provide healthier living and greener environments while delivering fast movements for customers. A few examples of BSSs are *Capital Bikeshare* in Washington DC, *Hubway* in Boston, *Bixi* in Montreal, *Vélib'* in Paris, *Wuhan and Hangzhou* Public Bicycle in China, etc.

Bike sharing systems are currently adopted in 1,139 cities with a fleet of over 1,445,000 bicycles. In addition, there are 357 cities where BSSs are either in the planning stage or under construction (Meddin & DeMaio, 2016). Figure (1) provides a quick view of the bike sharing systems around the world. In a typical bike sharing system, a set of base stations is strategically placed throughout the city. At the beginning of the day, each station is stocked with a pre-determined number of bikes. Users can hire and return bikes from any designated station, each of which has a finite number of docks. Many bike sharing operators use vehicles (e.g., trucks) to reposition bikes at the end of the work day so as to return to a pre-determined configuration. In addition, several bike sharing operators (e.g., *Capital Bikeshare* in Washington DC, *Hubway* in Boston) reposition the bikes during the day using myopic and adhoc methods.

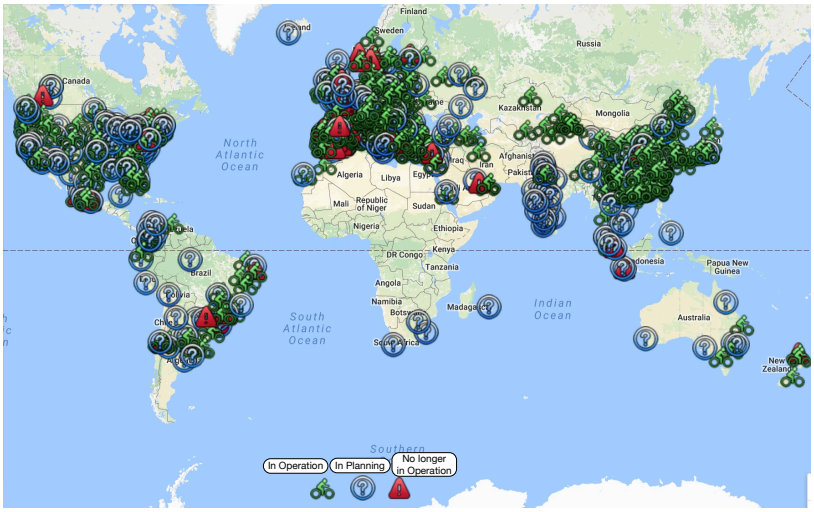


Figure 1: Visualisation of the BSSs worldwide (Meddin & DeMaio, 2016).

Due to the individualistic movements of customers according to their personal needs, there is often congestion (more than required) or starvation (fewer than required) of bikes at base stations. Figure (2) provides the number of instances[1] when stations were empty or full throughout various months in 2013-2014 for *Capital Bikeshare*. 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 were approximately 100 cases of empty stations and 100 cases of full stations per day. At a maximum, there were approximately 750 cases of empty stations and 330 cases of full stations per day. Moreover, in around 40% of instances, the stations were empty or full for more than 30 minutes. In order to tackle this problem, we employ dynamic repositioning of bikes during the day to better match demand with supply. Dynamic repositioning refers to considering movements of bikes by customers (which are usually significant and not negligible) during the repositioning period (Shu, Chou, Liu, Teo, & Wang, 2013). On the other hand, static repositioning refers to ignoring movements of bikes by customers during the rebalancing process (Chemla, Meunier, & Wolfler Calvo, 2013).

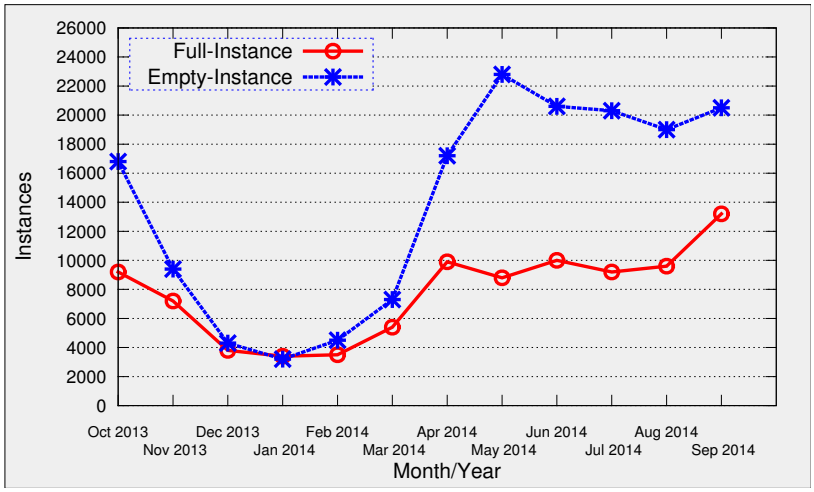


Figure 2: Number of empty and full instances of stations in *Capital Bikeshare*.

1. The data is taken from *Capital Bikeshare* [http://cabidashboard.ddot.dc.gov/cabidashboard/#Home].

As demonstrated by Fricker and Gast (2016) and our experimental results, starvation/congestion can result in a significant loss in customer demand. Such loss in demand can have two undesirable outcomes: (1) loss in revenue, and (2) increase in carbon emissions, as people can resort to fuel burning modes of transport. Moreover, some operators (e.g., *Vélib'* in Paris) are even penalised by the local government for loss in customer demand (Schuijbroek, Hampshire, & Van Hoeve, 2017). So, there is a practical need to minimise the lost demand and our approach is to dynamically reposition bikes with the help of vehicles while considering future expected demand extracted from past data. Furthermore, to ensure that the minimisation in demand loss is commercially viable, we consider an objective that is a trade-off between minimising lost demand (alternatively maximising profit) and minimising cost incurred by vehicles.

We refer to the joint problem of bike repositioning and vehicle routing as the Dynamic Repositioning and Routing Problem (DRRP). The DRRP with minor modifications can be used to represent the problem of repositioning empty cars in car sharing systems (e.g., Car2go, Zipcar) (Kek, Cheu, Meng, & Fung, 2009; Barth, Todd, & Xue, 2004), empty vehicles in Personal Rapid Transit (Lees-Miller, Hammersley, & Wilson, 2010) and idle ambulances in emergency response (Yue, Marla, & Krishnan, 2012; Saisubramanian, Varakantham, & Chuin, 2015; Ghosh & Varakantham, 2016). For example, in the case of car sharing, there is a need to continuously reposition cars to different parking spaces during the day to match with the pattern of demand.

Given the benefits of BSSs and the challenges of setting up such systems to operate efficiently, there have been a wide spectrum of research papers addressing the problem of lost demand and other issues pertinent to it. We have referred to these papers in Section 2. Some of the major differences between this paper and existing research are as follows: (1) we generate routing and repositioning decisions for multiple time steps (e.g., multiple periods during an entire day) using expected demand for bikes at each base station and at each time step, and (2) we employ novel approaches based on decomposition and abstraction to provide scalable solutions to DRRPs for large scale BSSs.

Due to a trivial reduction from the existing Static Bicycle Repositioning Problem (SBRP) which is NP-Hard (Schuijbroek et al., 2017), the DRRP is at least NP-Hard. Therefore, we focus on developing principled approximation methods. Our key contributions are as follows:

1. A mixed integer linear program (MILP) formulation to maximise profit for the BSS that considers the trade-off between:
   - maximising served demand, and
   - minimising cost incurred by vehicles.
2. A dual decomposition mechanism to decompose the MILP into two components – one which computes a repositioning solution for bikes and one which computes a routing solution for vehicles.
3. An abstraction mechanism that clusters the base stations in proximity to reduce the size of the problem and further speed up the solution process.
4. Extensive computational results using a simulation based on the real-world data sets of two bike sharing systems, namely, *Capital Bikeshare* (Washington, DC) and *Hubway* (Boston, MA), which demonstrate that our techniques can significantly reduce lost demand and improve operational efficiency of BSSs.

We describe the relevant work in Section 2 and the Dynamic Repositioning and Routing Problem (DRRP) model in Section 3. We then explain the learning of the DRRP model from a given data set in Section 3.1 and the approach for solving the resulting DRRP model in Sections 4, 5 and 6. Experimental setup and results to verify the utility of our approach are described in Sections 7 and 8 respectively. Finally, we conclude with a detailed discussion on resolving issues involved in taking our approach to a real bike sharing system in Section 9.

2. Related Work

Given the practical benefits of bike sharing systems, they have been studied extensively in the literature. In a recent survey, Laporte, Meunier, and Wolfler Calvo (2015) summarise the leading contributions in shared

transportation systems in terms of strategic planning (e.g., location of stations and sizes of fleet) and operational planning (e.g., vehicle repositioning). To situate these contributions in the context of this paper, we summarise along four threads and also compactly show them in Table (1). These four threads are described in the first four subsections of this section. Furthermore, the broad problem we are considering is one of sequential decision making in the presence of uncertainty. This problem is considered extensively in the Artificial Intelligence (AI) literature through Markov decision processes (MDPs) and its variants, stochastic network design in sustainability applications, and others. Thus, in the last subsection, we summarise other research relevant to our methodological contributions, i.e., decomposition and abstraction in sequential decision making under uncertainty.

Paper(s)	Online / Offline	Static (S)/ Dynamic (D)	Routing	Repositioning	Multi-step matching
Schuijbroek et al. (2017)	Online	S	✓	✓	✗
Raviv and Kolka (2013); Raviv et al. (2013)	Offline	S	✓	✗	✗
Raidl et al. (2013)	Offline	S	✓	✗	✗
Shu et al. (2010, 2013)	Offline	D	✗	✓	✓
Chemla et al. (2013)	Offline	S	✓	✗	✗
Contardo et al. (2012)	Online	D	✓	✓	✗
Nair et al. (2013); Nair and Miller-Hooks (2011)	Online	D	✗	✓	✗
O'Mahony and Shmoys (2015)	Offline	D	✓	✗	✗
Pfrommer et al. (2014); Singla et al. (2015)	Online	D	✗	✓	✗
Our approach (DRRP)	Offline	D	✓	✓	✓

Table 1: Summary of the related literature.

2.1 Designing a BSS

The first thread of research focuses on how to design a BSS. Kumar and Bierlaire (2012) provide a linear regression model to learn the correlation between station locations and customer's locations. They use customer demographics and personal information to identify the best locations for the placement of base stations. Martinez, Caetano, Eiró, and Cruz (2012) provide an MILP formulation to optimally locate the stations, and solve it using a branch and bound algorithm. Lin and Yang (2011) and Lin, Yang, and Chang (2013) propose a decision support model to design a BSS. In addition, they provide a model that incorporates the service level requirements of BSS by employing constraints from the inventory management literature.

The focus in this thread of research is on strategic planning and hence is complementary to the focus in this paper, which is on operational planning (day-to-day operations).

2.2 Static Repositioning

The second thread of research focuses on the rebalancing problem in static case where the movements of bikes during the repositioning period are negligible. Specifically, it focuses on the problem of finding routes for a fixed set of vehicles to reposition bikes at the end of the day or when the movements of bikes by customers are insignificant to achieve the desired configuration of bikes across the base stations. This problem is also known as the Static Bicycle Repositioning Problem (SBRP).

Benchimol, Benchimol, Chappert, De La Taille, Laroche, Meunier, and Robinet (2011) present an approximate solution (inspired by the solution of *C-delivery TSP*) to solve the static rebalancing and routing problem with a single vehicle. In a similar direction, Chemla et al. (2013) solve the static rebalancing problem for a single vehicle by employing a branch and cut algorithm that can solve a large scale problem with more than a hundred stations. However, in the presence of multiple vehicles, these solution approaches are not very effective as the routing solution of the first vehicle controls the routing solution of other vehicles and it affects the overall solution quality. Raviv and Kolka (2013) and Raviv, Tzur, and Forma (2013) address this concern and provide scalable exact and approximate solution approaches by introducing a set of MILP formulations to solve the SBRP with multiple vehicles. By employing an objective function that penalises unavailability of bikes or empty docks, they find solutions for the vehicles to reposition the bikes in a static manner. Raidl, Hu, Rainer-Harbach, and Papazek (2013), Di Gaspero, Rendl, and Urli (2013), and Di Gaspero, Rendl, and Urli (2016) propose approximate solutions for solving the SBRP using variable neighbourhood search based heuristics. Erdoğan, Laporte, and Wolfler Calvo (2014) present an integer programming formulation to solve the SBRP with demand intervals for a single vehicle. This is an empirically harder problem than the SBRP and they provide a Benders decomposition scheme for solving the problem efficiently. Unfortunately these solutions are not suitable for solving the DRRPs, as the movements of bikes during the planning period make the static solutions irrelevant[2].

Schuijbroek et al. (2017) propose a scalable approximate solution for the SBRP. They cluster base stations using a maximum spanning star (MAXSPS) approximation and allocate one vehicle to each of the clusters so that the service level requirements are satisfied. In addition, they represent this problem as a clustered vehicle routing problem (Battarra, Erdoğan, & Vigo, 2014). They assume that the movements of bikes by customers during the repositioning period are negligible. However, an online version of their approach can easily be employed to solve the dynamic repositioning problem using a rolling horizon framework that generates a routing and repositioning solution for each time step. In fact, we provide a comparison of our approach against their approach, which can be considered as a representative of the adhoc and myopic heuristic, in the experimental results section. As demonstrated in our experimental results, such a myopic repositioning solution can significantly falter as it does not consider the future demand surges.

Our approach differs from approaches mentioned in this section, as we consider the repositioning of bikes in the presence of bike movements by customers during the rebalancing period. Such bike movements can make planned static repositioning irrelevant when customers return the required number of bikes to a station or counterproductive when vehicles pick up bikes from a station where customers need bikes. In addition, a static solution typically cannot adequately capture the surges in customer demand even if they are predictable.

2.3 Dynamic Repositioning

Since service level requirements in a BSS change over time due to involuntary movements of customers, static solutions can significantly falter when employed for performing operational planning in BSS. Thus, the third thread of research focuses on dynamically repositioning bikes based on the assessment of demand.

Pfrommer, Warrington, Schildbach, and Morari (2014) provide a myopic repositioning and routing solution for individual vehicles based on the assessment of demand for the next 30 minutes. In case of multiple vehicles, they employ a greedy approach, where the solutions for the vehicles are determined sequentially one at a time. Furthermore, an additional incentive is given to the customers if they return their bikes to the neighbouring starving station rather than submitting it to their original destination station. Along a similar

2. Static solutions can either pick up bikes from a station where there is demand or add bikes to a station where there is no demand, as they do not consider the current context of bike inventory in conjunction with past patterns of demand.

direction, Singla, Santoni, Bartók, Mukerji, Meenen, and Krause (2015) propose a pricing mechanism that dynamically calculates the incentive values for each neighbouring station and offers the corresponding incentive amounts to the users through a mobile application. Contardo, Morency, and Rousseau (2012) propose a dynamic repositioning model to deal with the loss in customer demand in rush hours. They provide a myopic repositioning solution by considering the current demand that was recently observed before the repositioning decisions are made. Furthermore, they employ Dantzig-Wolfe and Benders decomposition techniques to speed up the solving process. However, due to the complex structure of the problem, there was a significant gap between their solution and its lower bound. In addition, due to the fact that these myopic solutions do not capture the future expected demand for multiple time steps in the planning phase, they usually provide poor quality solutions when the demand is dynamic (as observed in both our data sets).

Shu, Chou, Liu, Teo, and Wang (2010) and Shu et al. (2013) overcome this issue by considering the future expected demand for the entire planning horizon. They predict the stochastic demand from user trip data of Singapore metro system using a Poisson distribution. They provide an optimisation model that suggests a set of locations for the stations and propose a dynamic repositioning model to minimise the number of unsatisfied customers. However, they assume that repositioning of bikes from one station to another is always possible without considering the routing of vehicles.

Our approach differs from this thread of research as we consider the dynamic repositioning of bikes in conjunction with the routing of all the vehicles. In addition, we also consider multi-step expected demand that can account for demand surges at later time steps.

2.4 Demand Prediction and Analysis

The fourth thread of research which is complementary to the research presented in this paper is on demand prediction and analysis. Borgnat, Abry, Flandrin, and Rouquier (2009) and Borgnat, Abry, Flandrin, Robardet, Rouquier, and Fleury (2011) propose the idea of predicting temporal user demand from the past data and providing forecasted information to users. Froehlich, Neumann, and Oliver (2008) and Lathia, Ahmed, and Capra (2012) predict the user demand in terms of available bikes or normalised available bikes in a station at a certain time period. O'Mahony and Shmoys (2015) predict the service level requirements for each station in rush hours by analysing the data of *Citibike* in New York City. Furthermore, they provide an optimisation model to minimise the maximum imbalance such that users are not too far from bikes or available docks. Specifically, all the above mentioned papers provide data-driven analysis of the system-wide demand in BSSs.

In contrast to the data-driven analysis of the demand, Nair and Miller-Hooks (2011) and Nair, Miller-Hooks, Hampshire, and Bušić (2013) provide a theoretical analysis of the service level of a BSS using dual-bounded joint-chance constraints. They predict the near future demand for a short period and ensure that the system-wide demand is served with a certain probability. While these insights are practical and useful in demand prediction, they are not applicable for large systems. Leurent (2012) represents the bike sharing system as a dual Markovian waiting system to predict the actual demand. In contrast, we assume that the users are impatient and leave the system if they encounter an empty station. Raviv et al. (2013) and Schuijbroek et al. (2017) represent a BSS as a queueing network with Markov assumptions. The pickup or drop-off of bikes by the users are represented as random variables with a known probability distribution. While these assumptions hold for one step or short term planning, it becomes intractable for multi-step or long-term planning due to the time-varying nature of the demand and the inter-dependency between the pickup and drop-off rates between stations at consecutive time steps.

Given its wide ranging applicability and accuracy, Poisson distributions have been extensively used in the literature to represent random arrival processes. It has also been used regularly to represent customer arrivals at base stations in BSSs (George & Xia, 2011; Shu et al., 2010, 2013). Due to its simplicity and accuracy in representing customer arrival processes, we also represent the demand arrival process at each of the stations at each time step as a Poisson distribution.

2.5 Abstraction and Decomposition in Sequential Decision Making Problems under Uncertainty

This last thread of research is relevant to our two main methodological contributions, namely, abstraction and dual decomposition. Both these general methods have been applied in various types of sequential decision making problems in the literature. Our work is focused on multi-step matching of demand and supply, which has similarities with well-studied models in sequential decision making. Since we consider optimising a reward, the relevant sequential decision making model would be Markov decision process (MDP).

In fact, by considering every possible match of supply and demand as a potential state and modifications to each state as actions, we can represent a DRRP as an MDP. The corresponding MDP can be represented using following tuple $\langle S, A, P, R \rangle$:

- The state space, S of the MDP needs to incorporate all the possible combinations of inventory levels of the stations. Furthermore, to capture the physical limitations of vehicle routes and the number of bikes they can pick up or drop off from their origin station, the possible inventory levels and locations of the vehicles need to be incorporated through the state representation.
- Any changes in the inventory state of the stations by the vehicles can be represented through the action space, A of the MDP.
- The transition probabilities, P can be represented as the prior probabilities of moving bikes by the customers at a given state, according to the expected customer demand.
- The reward, R can be represented as a function of a given state at a time step and the expected customer demand at that time step.

However, since the number of possible matches grows exponentially with the number of stations, bikes and vehicles, the number of possible matches is extremely large[3]. Thus, it would be difficult to even specify the problem. Even if we are able to specify the problem, abstraction of state space for the MDP of a corresponding DRRP will automatically abstract the action space. As outlined in the next two paragraphs, existing abstraction techniques only consider state abstraction where action space remains the same, so they cannot be employed to solve the large scale MDP representations of DRRPs.

State abstraction has been widely adopted in Artificial Intelligence (Giunchiglia & Walsh, 1992) and Operations Research (Rogers, Plante, Wong, & Evans, 1991). Spatio-temporal abstraction (Sutton, Precup, & Singh, 1999; Mahadevan, 2002) is studied heavily in reinforcement learning (RL) literature, which is also a sequential decision making problem under uncertainty. Furthermore, abstracting time periods is widely used in the inventory routing problem (IRP) literature in order to simultaneously take into account the inventory planning and vehicle routing constraints (e.g., see Coelho, Cordeau, & Laporte, 2013; Papageorgiou, Nemhauser, Sokol, Cheon, & Keha, 2014).

Li, Walsh, and Littman (2006) propose five methods to perform state aggregation in MDPs while preserving the useful information that is critical for solving the complete problem. However, they assume that the action set does not change after abstraction, so the actions and policies of the original MDP and the abstract MDP are comparable. On the contrary, both the actions and states (when the DRRP is represented as an MDP) would change in our case, as the actions correspond to moving vehicles to a base station and the number of base stations changes with abstraction. Therefore, the insights of lossless abstraction are not applicable in our case. Given the extremely large scale, we have to abstract at the level of state features and not at the level of individual states, so that is another differentiating factor from the work by Li et al. (2006). Abstracting a relevant set of features for state abstraction has been employed in both the offline (Sturtevant & White, 2006) and online (Geramifard, Doshi, Redding, Roy, & How, 2011) settings for solving MDP and RL problems. These approaches focus on function approximation based on abstract state features and assume that the action space remains the same after abstraction. However, in our case, the abstraction of state features will also alter the action space. Due to this reason, these existing approaches cannot be directly applied for solving the DRRP.

3. For *Capital Bikeshare*, we have 300 stations, each with a capacity of 20 and 5 vehicles, each with a capacity of 20, so there would be approximately $20^{300} \times 20^5 \times 300^5$ states and $(300 \times 20)^5$ actions.

Furthermore, these existing approaches are suitable for problems with either low dimensional state space or with binary features. However, an MDP corresponding to a DRRP will have a large feature set and each feature can have a large number of values. As a consequence, the existing approaches will not be scalable for solving the DRRP.

Abstraction employed in this paper is similar to the work of Knoblock (1993). Specifically, their method (Knoblock, 1991) consists of three steps: (1) identify the abstract problem from the original problem, (2) solve the abstract problem, and (3) use the solution from the abstract problem to determine the solution for the original problem. The abstraction mechanism we employ in this paper to group the base stations is inspired by the recent work of Lu and Boutilier (2015). They use an abstraction technique to solve the multi-campaign, multi-channel marketing optimisation problem (MMMOP) by dynamically segmenting the customer population into a number of groups and use the solution of the abstract problem to guide the global solution. In a similar way, we group the nearby base stations into an abstract station because the customers using those stations have similar movement dynamics in aggregation and nearby stations have similar patterns of imbalance as shown in Section 6.

Dual decomposition has been employed in the literature to speed up the solution of sequential decision making problems. Dean and Lin (1995) use a decomposition technique to solve large scale MDPs. They decompose the original problem into a number of subproblems which are solved independently and finally combine the solutions to determine the solution of the original problem. Furmston and Barber (2011) employ Lagrangian dual decomposition to decompose a stationary policy finite-horizon MDP into a series of unconstrained MDPs that can be solved easily and use these solutions to guide the solution of the master problem in an iterative fashion. Recently, dual decomposition techniques have been employed successfully for solving sequential decision making problems such as factored MDPs (Guestrin & Gordon, 2002), spatial conservation planning (Kumar, Wu, & Zilberstein, 2012) and contact center planning (Kumar, Singh, Gupta, & Parija, 2014) problems. In this paper, we adopt a dual decomposition framework to exploit the weak dependency between two critical components of the problem, namely, repositioning of bikes and routing of vehicles.

3. Model: DRRP

We now formally describe the generic model, Dynamic Repositioning and Routing Problem (DRRP), that can be used to represent problems like the ones arising in BSSs. We employ 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, where each station $s \in \mathcal{S}$ has a fixed capacity (the number of docks) denoted by $C_s^{\#}$.
- $\mathcal{V}$ represents the set of vehicles that can be employed to reposition bikes and each vehicle $v \in \mathcal{V}$ has a fixed capacity (e.g., the number of slots for bikes) denoted by C_v^{*}.
- Initial distribution of bikes at base station s is given by $d_s^{\#,0}$. The number of bikes present in a vehicle v at the beginning of the day is given by $d_v^{*,0}$.
- $\sigma_{v,s}^0$ is set to 1 if vehicle v is stationed at station s initially. For ease of notation, we use the generic $\sigma_{v,s}^t$ and set it to 0, if $t > 0$.
- $F_{s,s'}^{t,k}$ represents the number of customers at time step t going out from station s and wanting to reach station s' at time step $t + k$.
- $R_{s,s'}^{t,k}$ represents the revenue obtained if a bike is hired at time step t from station s and is returned at station s' after k time steps. In a general case, the revenue depends only on the duration k, but we use the notation to represent a generic model[4]. We further note that this revenue parameter can also be used

4. Such a generic model can for instance be used to capture higher price of hiring bikes in central business districts.

to represent the monetary value of the social and environmental benefits associated with the usage of bike.

- $P_{s,s'}$ represents the routing cost for any vehicle to travel from station s to s' which depends on the distance between the two stations and the fuel cost. This cost parameter is then multiplied by a predefined factor to incorporate other relevant operating costs.

The goal is to maximise the overall profit. It should be noted that we do not directly minimise the lost demand because that can result in a significant cost of the routing of vehicles. This profit measure provides a trade-off between minimising lost demand and reducing routing cost of vehicles. The DRRP can be considered as a generalisation of the static bicycle repositioning problem (SBRP) (Schuijbroek et al., 2017). The SBRP in turn can be reduced from the 1-PDTSP problem, which is a known NP-Hard problem (Hernández-Pérez & Salazar-González, 2004). Due to this relation, it can be shown that the DRRP is at least NP-Hard.

3.1 Populating DRRP from Data of Bike Sharing Systems

The details of the data items provided by the bike sharing data sets (of two real BSSs), which we use in this paper are mentioned in Table (2).

Data Item	Definition
D1	The identification numbers, locations, capacities of base stations and the number of bikes present at each base station at the beginning of the day.
D2	Total number of vehicles available for repositioning and capacities of those vehicles.
D3	Customer trip history records.
D4	Revenues associated with successful customer bookings.

Table 2: Definition of the data items provided in real-world data sets.

From these data items, we populate the DRRP tuple $\left\langle \mathcal{S}, \mathcal{V}, \mathbf{C}^{\#}, \mathbf{C}^{*}, \mathbf{d}^{\#,0}, \mathbf{d}^{*,0}, \{\sigma_v^0\}, \mathbf{F}, \mathbf{R}, \mathbf{P} \right\rangle$ as follows:

- Set of base stations, $\mathcal{S}$ and their capacities, $\mathbf{C}^{\#}$ are obtained from data item **D1**.
- Set of vehicles, $\mathcal{V}$ and their capacities, $\mathbf{C}^{*}$ are obtained from data item **D2**.
- $\mathbf{d}^{\#,0}$ is obtained from data item **D1** and $\mathbf{d}^{*,0}$ is set to $\mathbf{0}$ for all vehicles (i.e., vehicles start out empty at the beginning of the day).
- We set the starting positions of vehicles, $\{\sigma_v^0\}$ randomly. Note that we have experimented with different starting configurations, but they do not have a major impact on the results. This is primarily because the positions of vehicles were changed based on the decisions to accommodate the flows of demands after the first time step.
- The demand and transition model, $\mathbf{F}$ is constructed by aggregating the customer trips for each time step over the data in **D3**. We aggregate the demand for each day of the week, i.e., there is a separate demand model for Mondays, Tuesdays, etc.
- Revenue model, $\mathbf{R}$ is constructed from the data item **D4**[5].
- Cost model, $\mathbf{P}$ is computed based on fuel costs in the location of the bike sharing system in conjunction with distances between base stations that are obtained from the data item **D1**.

In Figure (3), we provide an example to better explain the DRRP.

5. Typically, the first 30 minutes for subscription rides is free and after that an additional charge is applied. In our model, to ensure consistency, we can represent revenue for first 30 minutes as the subscription fee divided by the average number of rides.

Example 3.1 *For ease of explanation, we take 3 base stations and the movements of bikes between stations are shown over 3 time steps. An oval represents a base station at a time step. The leftmost oval at the top is base station 1 at time step 1, the oval in the middle column at the top is base 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 number in the square box on top of each oval represents the actual demand in that station at that time step. The number on each arc shows the actual flow of bikes on that arc. The blue ellipse on the top of an oval represents the lost demand at that station due to unavailability of bikes. For this specific example, the total loss in demand is 4 as shown in Figure* (3a)*. However, if we reposition the idle bikes efficiently with the help of a vehicle (in Figure* (3b)*, the routes for the vehicle are shown using a dotted line and the reposition numbers are shown within the circle associated with each dotted line), then there is no lost demand.*

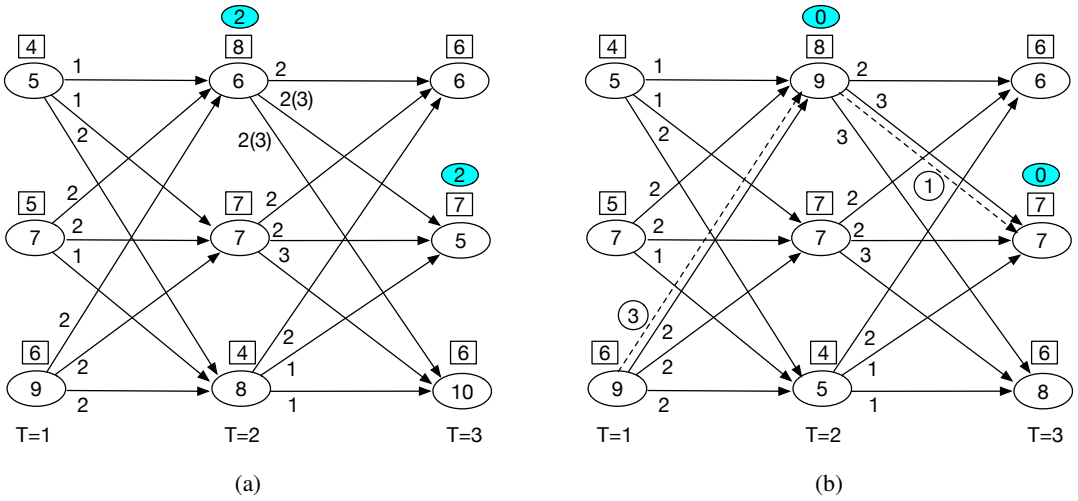


Figure 3: An example to explain the need for dynamic repositioning: (a) without repositioning (lost demand = 4), (b) with repositioning (dotted line represents the repositioning solution, lost demand = 0).

4. Optimisation Model for Solving DRRP

In this section, we provide an optimisation model for a given DRRP. Specifically, we provide a mixed integer linear program (MILP) that computes a profit maximising repositioning and routing solution. For ease of understanding, the decision and intermediate variables employed in the formulation are described in Table (3).

Category	Variable	Definition
Decision	$y_{s,v}^{+,t}$	The number of bikes picked from station s by vehicle v at time step t.
	$y_{s,v}^{-,t}$	The number of bikes dropped at station s by vehicle v at time step t.
	$z_{s,s',v}^{t}$	Set to 1 if vehicle v moves from station s to s' at time step t.
Intermediate	$x_{s,s'}^{t,k}$	The number of bikes moving from station s at time step t to s' at $t+k$ by the customers.
	$d_{s}^{\#,t}$	The number of bikes present in station s at time step t.
	$d_{v}^{*,t}$	The number of bikes present in vehicle v at time step t.

Table 3: Decision and intermediate variables.

The MILP for solving the DRRP is presented compactly in Table (4). Intuitively, the constraints in the optimisation model ensure that: the flows of bikes in and out of the base stations and vehicles are preserved

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

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

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

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

$$\sum_{s' \in \mathcal{S}} z^t_{s,s',v} - \sum_{s' \in \mathcal{S}} z^{t-1}_{s',s,v} = \sigma^t_{v,s}, \quad \forall t, s, v \quad (5)$$

$$\sum_{s' \in \mathcal{S}, v \in \mathcal{V}} z^t_{s,s',v} \leq 1, \quad \forall t, s \quad (6)$$

$$y^{+,t}_{s,v} + y^{-,t}_{s,v} \leq C^*_v \cdot \sum_{i \in \mathcal{S}} z^t_{s,i,v}, \quad \forall t, s, v \quad (7)$$

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

$$z^t_{s,s',v} \in \{0,1\} \quad (9)$$

Table 4: SOLVEDRRP()

(constraints (2) and (4)), the flows of vehicles in and out of the base stations are preserved (constraints (5) and (6)), and capacities of base stations and vehicles are not violated (constraints (7) and (8)). More importantly, these constraints ensure that the flows of bikes between stations follow the flows of bikes observed in the demand and transition model, **F**. More specific details of the constraints employed in SOLVEDRRP() of Table (4) are as follows:

Objective: To represent the trade-off between lost demand (or alternatively the revenue from customer trips) and the cost of using vehicles, we employ the dollar value of both quantities and combine them into the overall profit. This objective is represented in Expression (1) of the MILP in SOLVEDRRP().

Flows of bikes in and out of stations are preserved: Constraints (2) enforce this flow preservation. Intuitively, in these constraints, we ensure that the number of bikes at a base station at a time step ($d^{\#,t+1}_s$) is equivalent to the sum of the number of bikes at the same base station in the previous time step ($d^{\#,t}_s$) and the net number of bikes coming into the station during that time step $\left(\sum_{k,\hat{s}} x^{t-k,k}_{\hat{s},s} - \sum_{k,s'} x^{t,k}_{s,s'} + \sum_v \left(y^{-,t}_{s,v} - y^{+,t}_{s,v}\right)\right)$.

Flows of bikes between any two stations follow the transition dynamics observed in the data: As a subset of arrival demand can be served if the number of bikes present in a station is less than the arrival demand, constraints (3) ensure that the flows of bikes between station s and s' should be less than the product of the 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 expected customer demand (i.e., $F^{t,k}_{s,s'} / \sum_{k,\hat{s}} F^{t,k}_{s,\hat{s}}$).

Flows of bikes in and out of vehicles are preserved: Constraints (4) enforce this flow preservation. Intuitively, in these constraints, we ensure that the number of bikes in a vehicle at a time step ($d^{*,t+1}_v$) is equivalent to the sum of the number of bikes in the vehicle at the previous time step ($d^{*,t}_v$) and the net number of bikes coming into the vehicle during that time step $\left(\sum_{s \in S} \left(y^{+,t}_{s,v} - y^{-,t}_{s,v}\right)\right)$.

Flows of vehicles in and out of stations are preserved: Constraints (5) enforce this flow preservation. Intuitively, in these constraints, we ensure that the number of vehicles going out of a station ($\sum_{s' \in \mathcal{S}} z^t_{s,s',v}$) is equivalent to the sum of the number of vehicles coming into the station ($\sum_{s' \in \mathcal{S}} z^{t-1}_{s',s,v}$) and the number of vehicles that were present at that station[6] ($\sigma^t_{v,s}$).

A maximum of one vehicle can be present in one station at any time step: Due to limited space availability near base stations and to avoid a synchronisation issue in pickup or drop-off events by multiple vehicles from the same station at the same time step, constraints (6) ensure that at any time step, the maximum number of vehicles at a station ($\sum_{s' \in \mathcal{S}} z^t_{s,s',v}$) is 1.

Vehicles can only pick up or drop off bikes at a station if they are present at that station: Constraints (7) enforce that the number of bikes picked up or dropped off from a station at each time step by each vehicle is bounded by whether the station is visited by the vehicle at that time step or not.

Station and vehicle capacities are not exceeded when repositioning bikes: Constraints (8) ensure that the number of bikes at a station s does not exceed the number of available docks at that station ($C^{\#}_s$). Similarly, these constraints also enforce that the number of bikes picked up or dropped off by a vehicle v in aggregate does not exceed the capacity of the vehicle (C^*_v).

The size of the above described model grows exponentially as the number of stations increases. To tackle this problem, we describe two mechanisms, namely, dual decomposition and abstraction to improve the scalability of the optimisation model delineated in Table (4).

5. Dual Decomposition Approach for Solving the 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 vehicles between base stations to pick up or drop off bikes) and the repositioning problem (how many bikes to pick up and drop off from each station). The following observation highlights this minimal dependency:

Observation 1 *In the MILP of Table* (4)*:*

- $\mathbf{y}^+$ *and* $\mathbf{y}^-$ *variables capture the solution for the repositioning problem.*
- $\mathbf{z}$ *variables capture the solution for the routing problem.*

These sets of variables only interact due to constraints (7)*. In all other constraints of the original model, the routing and repositioning variables are completely independent.*

In order to exploit observation (1), we use the well-known Lagrangian dual decomposition (LDD) (Fisher, 1985; Gordon, Varakantham, Yeoh, Lau, Aravamudhan, & Cheng, 2012) technique. While this is a general purpose approach, its scalability, usability and utility depend significantly on the following characteristics of the model:

1. *Identifying the right constraints to be dualized:* This step is crucial to ensure that the resulting subproblems are easy to solve and the resulting bound derived from the dual solution is tight during the LDD process. If the right constraints are not dualized, then the underlying Lagrangian based optimisation may not be decomposable or it may take significantly more time than the original MILP to find the desired solution.
2. *Extraction of a primal solution from an infeasible dual solution:* The primal extraction process is important to derive a valid bound (heuristic solution) during the LDD process. In many cases, the solution

6. This second term is greater than zero for the first time step only and for rest of the time steps it is set to zero.

obtained by solving the decomposed dual slaves can be infeasible with respect to the original formulation and hence, the overall approach can potentially lead to slower convergence and poor solutions.

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$(\alpha^{it}, drrp)$
$\alpha^{it+1}_{s,t,v} \leftarrow \left[\alpha^{it}_{s,t,v} + \gamma \cdot (y^{+,t}_{s,v} + y^{-,t}_{s,v} - C^*_v \cdot \sum_i z^t_{s,i,v})\right]_+$
$p, x_p, y^-_p, y^+_p \leftarrow$ EXTRACTPRIMAL $(\mathbf{z}, drrp)$
$it \leftarrow it + 1$
until $\left[p - (o_1 + o_2)\right] \leq \delta$
return p, x_p, y^+_p, y^-_p, z

The pseudo code for the LDD is provided in Algorithm (1). We first decompose the original problem into a master problem and two slaves (SOLVEREDEPLOY() and SOLVEROUTING()). As highlighted in observation (1), only constraints (7) contain a dependency between routing and repositioning variables. Thus, we dualize constraints (7) using the dual variables, $\alpha_{s,t,v}$ and obtain the Lagrangian function (expressed as a minimisation problem as shown by Fisher, 1985) as follows:

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

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

In Equation (11), the first two terms correspond to the repositioning problem and the last term corresponds to the routing problem. The two subproblems corresponding to the repositioning and routing problems are given in Table (5) and Table (6), 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 5: SOLVEREDEPLOY()

From Equation (11), given an α, the dual value corresponding to the original problem is obtained by adding up the objective function values from the two slaves, which yields a valid lower bound with respect to the original problem. It should be noted that the decomposition is only for $\mathcal{L}(\boldsymbol{\alpha})$. Next, we have to solve the following optimisation problem at the **master** in order to reduce violations of the dualized constraints:

$$\max_{\boldsymbol{\alpha} \geq 0} \mathcal{L}(\boldsymbol{\alpha}) \quad (12)$$

This **master** optimisation problem is solved iteratively using a sub-gradient descent method applied on the dual variables α:

$$\alpha^{k+1}_{s,t,v} = \big[\alpha^k_{s,t,v} + \gamma \cdot (y^{+,t}_{s,v} + y^{-,t}_{s,v} - C^*_v \cdot \sum_i z^t_{s,i,v})\big]_+ \quad (13)$$

where the $[]_+$ notation indicates that the value must be equal or greater than zero. This is because we have dualized a "less than or equal to" constraint and a value of less than zero indicates that there is no violation of

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

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

Table 6: SOLVEROUTING()

the constraint. γ is a step-size parameter that is set using the standard strategy presented by Bertsekas (1999) (refer to section 6.3.1). The value within parenthesis () in Equation (13) is the sub-gradient and is computed from the solutions of the two slaves.

The algorithm terminates when the difference between the primal objective (defined as p in Algorithm 1) and the dual objective (the sum of the slave's objectives o_1, o_2) is less than a pre-determined threshold value δ. In order to compute the optimality gap[7], we need the best primal solution in conjunction with the dual solution. Therefore, it is important to obtain a primal solution after each iteration from the solutions of the slaves. In our case, however, the aggregate solution obtained from slaves may not always be feasible with respect to the original problem in Table (4).

Observation 2 *The infeasibility in the dual solution arises because the routes of the vehicles (obtained by solving the routing slave) may not be consistent with the repositioning plan of bikes (obtained by solving the repositioning slave). However, the solution for the routing slave is always feasible and can be fixed to obtain a feasible primal solution with respect to the original problem.*

Let, $Z^t_{s,v} = \sum_{s'} z^t_{s,s',v}$. We extract the primal solution by solving the optimisation formulation provided in Table (7). Essentially, we solve the repositioning slave with an additional set of constraints (14), which ensure that repositioning in a station is possible if a vehicle is present there. More specifically, constraints (14) are equivalent to constraints (7) where we use the solution values of the routing slave ($\mathbf{z}$) as the input. Thus, ExtractPrimal() satisfies all the constraints of Table (4) and produces a feasible solution to the original problem. Finally, we subtract the routing cost from the objective value to get the correct 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*

$$y^{+,t}_{s,v} + y^{-,t}_{s,v} \leq C^*_v \cdot Z^t_{s,v}, \qquad \forall t, s, v \qquad (14)$$

Table 7: EXTRACTPRIMAL()

Proposition 1 *(Fisher, 1985) : The error in the solution quality obtained by the Lagrangian dual decomposition method in Algorithm* (1) *is bounded by the difference between the primal objective, p and the dual objective, $(o_1 + o_2)$.*

6. Abstraction Approach for Solving DRRP

Even after applying the LDD, we can only solve problems with at most 60 base stations, 38 time steps and 5 vehicles within a threshold time of 12 hours. However, in most of the cities, the number of base stations is

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

higher. In this section, we provide an abstraction mechanism to further speed up the solution process. We first provide two observations from the real data that assist with deciding on the method to abstract base stations:

- Figure (4) provides heat maps of empty stations[8] during various times of the day for *Capital Bikeshare*. Figure (4a) and (4b) show the heat maps of empty stations in the morning peak hours. Similarly, Figure (4c) and (4d) show the heat maps of empty stations during the evening peak hours. All the heat maps indicate that the stations near to each other exhibit similar behaviour.
- Base stations are relatively close to each other. For instance, in case of the *Capital Bikeshare*, there are 5-6 base stations within 2 blocks (up to 0.4 miles or 0.64 kilometers) in the center of the city.

Figure 4: Heat maps for empty stations (data set: *Capital Bikeshare*): (a) morning peak (7AM - 9AM), (b) morning peak (9AM - 11AM), (c) evening peak (4PM - 6PM), (d) evening peak (6PM - 8PM).

From the above observations, since nearby stations exhibit similar behaviour and are also close enough to be covered by a carrier vehicle with minimal travel, we exploit the geographical proximity based clustering method to obtain abstract stations. Specifically, we employ relative distances between stations while

8. A station is considered empty if there are no bikes in the station for more than 2 minutes. For a specific time of a specific day (Monday, Tuesday, etc.), we use the trip history data and count the number of times each station became empty over a reasonably long duration (a year). Red corresponds to stations that became empty frequently, green corresponds to stations that became empty moderately and purple is for stations that rarely became empty.

clustering stations into abstract stations. We follow the following three steps typically employed in abstraction and introduced by Knoblock (1991): (1) create an instance of the DRRP with abstract stations, each of which is a group (obtained from clustering) of the original base stations, (2) solve the abstract DRRP using the LDD and obtain the routing and repositioning solution over abstract stations, and (3) derive the routing and repositioning solution for the original DRRP from the routing and repositioning solution of the abstract DRRP.

6.1 Create Abstract DRRP

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

- Stations in the abstract DRRP, $\tilde{\mathcal{S}}$: Grouping of stations $\mathcal{S}$ into abstract stations can either be done by an expert or computed by a clustering approach[9] (e.g., *k*-means clustering). Thus, each abstract station $\tilde{s} \in \tilde{\mathcal{S}}$ 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 an abstract station $\tilde{s}$ is the sum of capacities of all the stations $s \in \tilde{s}$.
- Initial distribution of bikes at the abstract station: $d_{\tilde{s}}^{\#,0} = \sum_{s \in \tilde{s}} d_s^{\#,0}$. The initial distribution of bikes of an abstract station $\tilde{s}$ is the sum of the initial distributions of all the stations $s \in \tilde{s}$.
- Initial distribution of vehicle: $\sigma_{v,\tilde{s}}^0 = 1$, if $\exists s \in \tilde{s}$, $\sigma_{v,s}^0 = 1$. The vehicle v is initially located in abstract station $\tilde{s}$ if its original location (i.e., station s) belongs to the abstract station $\tilde{s}$.
- Flows of bikes in the abstract DRRP: $F_{\tilde{s},\tilde{s}'}^{t,k} = \sum_{\{s \in \tilde{s}, s' \in \tilde{s}'\}} F_{s,s'}^{t,k}$. The flows of bikes from an abstract station $\tilde{s}$ to $\tilde{s}'$ are calculated as the sum of the flows of all the bikes taken by the customers from any station $s \in \tilde{s}$ to a station $s' \in \tilde{s}'$ in the original DRRP.
- Routing cost for the vehicles in the abstract DRRP: $P_{\tilde{s},\tilde{s}'} = \max_{\{s \in \tilde{s}, s' \in \tilde{s}'\}} P_{s,s'}$. We consider a conservative option of taking the worst case penalty. Specifically, we take the maximum routing cost for traveling between any pair of stations $s \in \tilde{s}$ and $s' \in \tilde{s}'$.

6.2 Solve the Abstract DRRP

In the second step, we use the LDD approach from Section 5 to solve the abstract DRRP. There are two possible assumptions we can make about the movements of vehicles in an abstract station: (1) a vehicle can visit all stations of an abstract station in a single time step, and (2) a vehicle can visit one station within an abstract station in a single time step.

6.2.1 VEHICLE CAN VISIT ALL STATIONS OF AN ABSTRACT STATION IN A SINGLE TIME STEP

As a vehicle can visit all the stations of an abstract station within one time step, we need to ensure that at most one vehicle is present in an abstract station in each time step to avoid the inconsistency in pickup or drop-off events by different vehicles. Therefore, the optimisation model for solving the abstract DRRP is equivalent to the one shown in Table (4) and we directly use the LDD approach from Section 5 to efficiently solve the abstract DRRP.

9. The grouping of base stations can be done in various ways and the results may vary for different problem instances. Clustering base stations according to geographical proximity is one option and the experimental results show that it provides a reasonable improvement over the two benchmark approaches.

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

s.t. *Constraints* (5) *&* (9) *hold*

$$\sum_{j \in \tilde{S}, v \in \mathcal{V}} \tilde{z}^t_{\tilde{s},j,v} \leq |\tilde{s}|, \qquad \forall t, \tilde{s} \quad (15)$$

Table 8: SOLVEABSTRACTROUTING()

6.2.2 VEHICLE CAN VISIT ONE STATION WITHIN AN ABSTRACT STATION IN A SINGLE TIME STEP

As the abstract stations contain multiple base stations, we need to modify constraints (6) to allow multiple vehicles in an abstract station. Table (8) provides the modified version of the routing slave to solve the abstract DRRP, where constraints (5) & (9) are defined over $\tilde{z}$. The modified set of constraints (15) ensure that at any time step maximum $|\tilde{s}|$ vehicles can visit an abstract station $\tilde{s}$. However, the repositioning slave and master function remain unchanged. There are two key outputs from the LDD algorithm: (1) repositioning solution, $\tilde{y}$ for moving bikes between abstract stations, and (2) routing solution, $\tilde{z}$ for moving vehicles between abstract stations at different time steps.

6.3 Deriving Solutions for the Original DRRP

In the third step, we retrieve the solution for the original DRRP from the abstract DRRP solution.

6.3.1 VEHICLE CAN VISIT ALL STATIONS OF AN ABSTRACT STATION IN A SINGLE TIME STEP

As we are abstracting the base stations based on their relative distance, all the base stations within an abstract station are located nearby (less than a few kilometers in our data sets). So, in reality it is possible for a vehicle to visit all the base stations of an abstract station within one time step. The mechanisms to retrieve the repositioning and routing solutions for the original DRRP from the abstract DRRP solution are outlined below.

Repositioning solution for the original DRRP: Based on a fixed routing solution, $\tilde{\mathbf{z}}$ for the abstract DRRP, we retrieve the repositioning solution for the original DRRP. Specifically, we fix the locations where vehicles will be present at different time steps and remove all constraints which are only related to vehicle routing (since the routing solution is fixed) in the optimisation model of Table (4). Solving this optimisation model yields a repositioning solution for the original DRRP. Formally, from the abstract DRRP solution $\tilde{\mathbf{z}}$, we obtain constants $\mathbf{Z}$ as follows:

$$Z^t_s = \begin{cases} 1, \textbf{if } s \in \tilde{s} \ \wedge \ \sum_{v,\tilde{s}'} \tilde{z}^t_{\tilde{s},\tilde{s}',v} = 1 \\ 0, \textbf{otherwise} \end{cases}$$

The final optimisation model to obtain the repositioning solution for the original DRRP is shown in Table (9). The key differentiating constraints that have not been used earlier are constraints (20). These constraints ensure that the 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 from the abstract station according to the repositioning solution of the abstract DRRP.

Routing solution for the original DRRP: Given the routing solution for the abstract DRRP (also referred to as the abstract routing solution), the vehicle assigned to each abstract station at a time step is fixed. From this abstract routing solution, our goal is to find the routing solution for all the stations within each abstract station at each time step. This routing solution must be consistent with the repositioning solution computed for the original DRRP. We use $\mathbf{Y}$ (instead of $\mathbf{y}$) to represent the final repositioning solution.

$$\max_{\mathbf{y}^+,\mathbf{y}^-} \sum_{t,s,s,s'} R^{t,k}_{s,s'} \cdot x^{t,k}_{s,s'} \tag{16}$$

$$\textbf{s.t.} \quad d^{\#,t}_s + \sum_{k,\hat{s}} x^{t-k,k}_{\hat{s},s} - \sum_{k,s'} x^{t,k}_{s,s'} + y^{-,t}_s - y^{+,t}_s = d^{\#,t+1}_s, \qquad \forall t, s \tag{17}$$

$$x^{t,k}_{s,s'} \leq d^{\#,t}_s \cdot \frac{F^{t,k}_{s,s'}}{\sum_{k,\hat{s}} F^{t,k}_{s,\hat{s}}}, \qquad \forall t, k, s, s' \tag{18}$$

$$y^{+,t}_s + y^{-,t}_s \leq C^*_v \cdot Z^t_s, \qquad \forall t, s \tag{19}$$

$$\sum_{s \in \tilde{s} | \sum_{\tilde{s}'} \tilde{z}^t_{\tilde{s},\tilde{s}',v}=1} [y^{+,t}_s - y^{-,t}_s] = d^{*,t+1}_v - d^{*,t}_v, \qquad \forall t, \tilde{s} \tag{20}$$

$$0 \leq x^{t,k}_{s,s'} \leq F^{t,k}_{s,s'}, y^{+,t}_s, y^{-,t}_s \leq C^*_v, d^{\#,t}_s \leq C^{\#}_s \tag{21}$$

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

For each vehicle, we compute the routing solution for the original DRRP incrementally by starting at the first time step and from the starting abstract station. We identify the route to be taken between all the base stations within this starting abstract station. Then, we move to the abstract station for the next time step recommended by the abstract routing solution and so on.

For each vehicle, the first step in computing the routing solution for stations within an abstract station is to identify the starting station[10]. We consider the starting station for non-starting abstract stations as the one that is nearest to the station from where the vehicle has exited in the previous time step. An advantage of this incremental method is that it minimises the routing cost for transition between abstract stations.

Once the starting station is obtained and the repositioning solution $\mathbf{Y}$ is known, we employ the optimisation model in Table (10) to find an intra-abstract station routing solution. We compute the best route within the stations of an abstract station $\tilde{s}$, while visiting each base station once and satisfying the repositioning numbers from each station, $\mathbf{Y}$.

$$\min_{\mathbf{z}} \sum_{t,s,s'} P_{s,s'} \cdot z^t_{s,s',v} \tag{22}$$

$$\textbf{s.t.} \quad \hat{d}^{*,t} + \sum_s (Y^+_s - Y^-_s) \cdot \sum_{s'} z^t_{s,s',v} = \hat{d}^{*,t+1}, \qquad \forall t \in \hat{T} \tag{23}$$

$$\sum_{t,s'} z^t_{s,s',v} = 1, \qquad \forall s \in \tilde{s} | (Y^+_s + Y^-_s) > 0 \tag{24}$$

$$\sum_{s'} z^t_{s,s',v} - \sum_{\hat{s}} z^{t-1}_{\hat{s},s,v} = \sigma^t_{v,s}, \qquad \forall t \in \hat{T}, s \in \tilde{s} \tag{25}$$

$$0 \leq \hat{d}^{*,t} \leq C^*_v, z^t_{s,s',v} \in \{0,1\} \tag{26}$$

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

The objective delineated in Expression (22) is to minimise the routing cost of the vehicle and the constraints are defined as follows:

10. Since the position of every vehicle is known at first time step in the original DRRP tuple, we have the starting station for the starting abstract station.

- *Flows of bikes in and out of a vehicle are preserved:* Constraints (23) enforce this by ensuring that the number of bikes in the vehicle at time step $t+1$ is equal to the sum of the number of bikes present in the vehicle at time step t plus the net number of bikes picked up from a station s at that time step. Note that, the time index t here is used to represent the sequence of moves for the vehicle between the base stations within an abstract station.
- *Each station is visited only once:* Constraints (24) restrict that each base station where a repositioning is required (i.e., $Y_s^+/Y_s^- > 0$) is visited only once.
- *Flow conservation of each vehicle at a station:* Constraints (25) ensure that the flow in to a station s (i.e., $\sum_{s'} z^{t-1}_{s',s,v}$) is equal to the flow out from that station at time step t (i.e., $\sum_{s'} z^{t}_{s,s',v}$). σ^0 represents the initial location of the vehicle and it is used to ensure that the vehicle moves appropriately out of the initial location.
- *Capacity of the vehicle is not exceeded during repositioning:* Constraints (26) ensure that the number of bikes picked up or dropped off by a vehicle in aggregate does not exceed the capacity of the vehicle (C_v^*).

Example 6.1 *Figure* (5) *provides a handcrafted toy example to illustrate the abstraction method where a vehicle can cover all stations within an abstract station in one time step. Solid dots represent the base stations and big circles represent the abstract stations. We considered 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 vehicle is indicated with a circle over the solid dot. Figure* (5a) *depicts the optimal abstract station level routing solution (by solving the LDD based global MILP on the abstract DRRP) for the vehicle. Figure* (5b) *depicts the base station level routing solution within the abstract station 1 at the 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 solution for the vehicle. Figure* (5c), (5d) *depict the base station level routing solution within abstract station 2 and 3 respectively.*

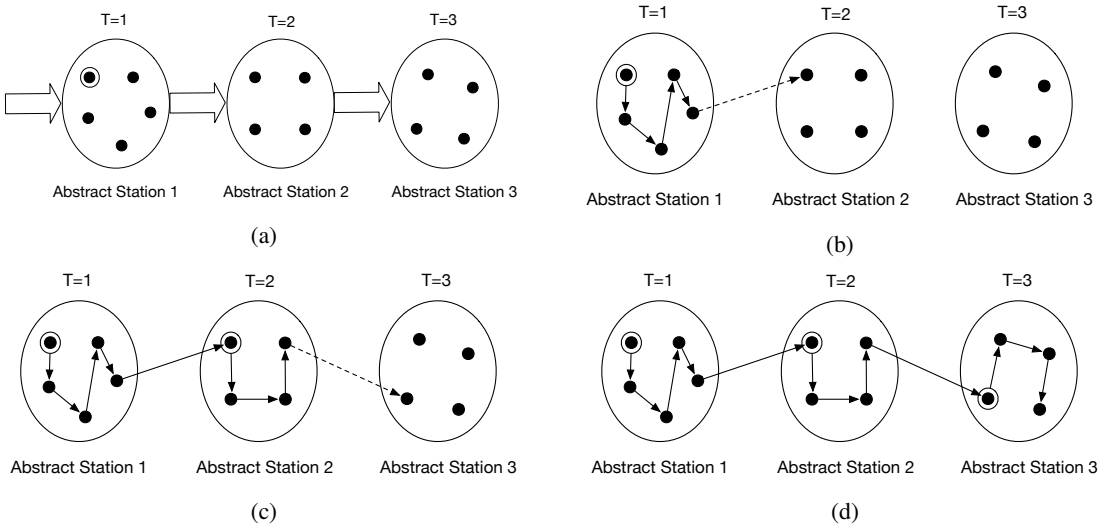


Figure 5: Routing solution in the abstract DRRP: (a) abstract station level routing solution, (b) routing solution within abstract station 1, (c) routing solution within abstract station 2, and (d) routing solution within abstract station 3.

6.3.2 VEHICLE CAN VISIT ONE STATION WITHIN AN ABSTRACT STATION IN A SINGLE TIME STEP

Table (11) provides the optimisation model to compute a feasible solution for the original DRRP with the assumption that a vehicle can only travel to one station in each time step. We solve the global MILP SolveDRRP() for the original DRRP provided in Table (4) with an additional set of constraints (27) to ensure that a vehicle can only be present in a base station at any time step if the station belongs to the abstract station where the vehicle is located in the abstract DRRP solution. Specifically, 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 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 easier to solve 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*

$$\sum_{s \in \tilde{s}, s' \in \tilde{s}'} z^t_{s,s',v} = \tilde{z}^t_{\tilde{s},\tilde{s}',v}, \qquad \forall \tilde{s}, \tilde{s}', t, v \quad (27)$$

Table 11: RETRIEVEDRRP()

6.4 Reasons for Improvement in Scalability

The scale of the optimisation models used for solving the abstract DRRP and deriving an original DRRP solution from the abstract DRRP solution are reduced in comparison to the original optimisation model. Hence, this abstraction method is able to substantially speed up the solution process. Specifically, here are the reasons for reduction in runtime of the optimisation models:

- **Reduction in the number of variables and constraints**: The number of variables and constraints in the optimisation model are significantly reduced. For instance, for a 300 station problem in the original optimisation model of Table (4), there would be 90000 binary decision variables, **z** for each time step. On the other hand, for an abstract DRRP with 50 abstract stations, there would only be 2500 **z** variables for each time step.
- **Relaxation**: Another important reason for significant improvement in scalability is that the optimisation models for computing routing and repositioning solution for the abstract DRRP are the relaxations of the optimisation model for the original DRRP. This is because constraints in the optimisation models for the abstract DRRP are obtained by aggregating the constraints that are present in the optimisation model for the original DRRP.

7. Experimental Setup

In this section, we describe the real and synthetic data sets that are used in the computational experiments, the benchmark approaches that are implemented for the computational comparisons, and the simulation model used to compute the comparison metrics.

Since our goal is to avoid people from going back to using private vehicles due to unavailability of bikes, the key comparison metric is the total amount of lost demand. To ensure that the amount of lost demand is reduced at no extra fuel cost to the operators, we also consider the total profit as a metric. The runtime is primarily employed to measure scalability and whether we are able to get a high quality solution within a reasonable amount of time.

7.1 Real and Synthetic Data Sets

We employ data sets of two leading bike sharing systems in US[11], namely, *Capital Bikeshare* (Washington, DC) and *Hubway* (Boston, MA), and the synthetic data sets are derived from these real-world data sets. The data items contained in the bike sharing data sets are previously mentioned in Table (2). In addition to the data items provided in the data sets, we collect data about the cost of fuel for vehicles[12] from authentic sources. It should be noted that we consider a significant overestimation of the costs to ensure our results are not too sensitive to these values. These elements of real-world data sets and the data collected from the authentic sources are used to populate the DRRP model.

As for the synthetic data sets, they are generated from the two real-world data sets as follows: (1) we take a subset of the stations from the real-world data sets, (2) customer demand, station capacity, geographical location of stations and initial distribution are drawn from the real-world data for the selected stations, and (3) we take the same revenue and cost model discussed earlier from the real-world data sets.

7.2 Approaches

We employ the commercial linear optimisation solver CPLEX to solve linear programs and mixed integer linear programs. We refer to the optimisation model of Table (4) as **MILP**. The dual decomposition method for solving the MILP that is described in Section 5 is referred as **LDD** (Lagrangian dual decomposition). Finally, we refer to the abstraction approach described in Section 6 as **Abstraction**. The overall approach (LDD+Abstraction) is referred to as **dynamic**.

The first approach that is used as a benchmark for performance comparison is the static repositioning approach (referred to as **static** in the graphs). This refers to the approach adopted by some of the bike sharing systems where bikes are repositioned only at the end of the day. We compare our approach with this current practice of not repositioning bikes during the day.

Through interactions with bike sharing operators, we understand that they typically employ myopic heuristics to reposition idle bikes. They detect the congestion or starvation of bikes in a station using sensors installed at the stations and reposition the bikes from base stations using myopic heuristics. To the best of our knowledge, we are not aware of any research paper that formally presents heuristics to solve the DRRP online. The method that can be employed online to reposition bikes can be adapted from the algorithm provided by Schuijbroek et al. (2017). This approach can be executed online with the assumption of negligible movements of bikes by customers (refer to Appendix A for the details of the approach and the changes made to ensure fair comparison). This approach is referred to as **online**.

7.3 Simulation Model

Similar to the work of Shu et al. (2013), we also evaluate the performance of relevant approaches by using a simulation that is based on the past data. There are two key aspects of the simulation that are important to note, namely, the impact of repositioning on transition dynamics of customers and the actual demand for bikes.

7.3.1 IMPACT OF REPOSITIONING ON TRANSITION DYNAMICS OF CUSTOMERS

Repositioning changes the number of bikes available in stations at different time steps in comparison to the observed data denoted by **F**. However, a reasonable assumption used for any configuration is that the aggregated transition probabilities between stations that are observed in the data remain the same. This assumption is also employed in previous work of Shu et al. (2013).

11. The data is taken from *Capital Bikeshare* [http://capitalbikeshare.com/system-data] and *Hubway* BSS [http://hubwaydatachallenge.org/trip-history-data].
12. The mileage results in Table 2 of Fishman, Washington, and Haworth (2014) show that carrier vehicles in BSS consume 1 litre of diesel for traveling approximately 12 kilometers. http://www.globalpetrolprices.com/diesel_prices/#USA shows that the price of diesel in January, 2017 is 0.67 USD per litre, but we overestimate it as 1.5 USD to include other operational costs.

The customer demand observed in the data, $F^{t,k}_{s,s'}$, denotes the number of bikes booked from station s at time step t and reached station s' at time step $t+k$. Based on the customer demand observed in the data set, this simulation generalises customer movements for a given number of bikes, $d^{\#,t}_s$ present in a station s at time step t[13]. Specifically, the actual movements of customers at each station are determined based on the following two cases: (1) if the total arrival demand at a station is less than the number of bikes present in the station, then all the customers are served, and (2) if total arrival demand at a station is higher than the number of bikes present at the station, then the actual flow (denoted as $x^{t,k}_{s,s'}$) depends on the ratio $\frac{F^{t,k}_{s,s'}}{\sum_{\tilde{s},k} F^{t,k}_{s,\tilde{s}}}$, i.e.,

$$x^{t,k}_{s,s'} = \left\{ \begin{array}{ll} F^{t,k}_{s,s'} & \textbf{if } \sum_{k,\tilde{s}} F^{t,k}_{s,\tilde{s}} \leq d^{\#,t}_s \\ \frac{F^{t,k}_{s,s'}}{\sum_{k,\tilde{s}} F^{t,k}_{s,\tilde{s}}} \cdot d^{\#,t}_s & \textbf{Otherwise} \end{array} \right\}. \tag{28}$$

We calculate the number of bikes present in a station at time step $t+1$ (see Equation 29) as the sum of the number of bikes left out in the station at time step t and the net incoming bikes in that time step.

$$d^{\#,t+1}_s = d^{\#,t}_s + \Big[\sum_{k,\tilde{s}} x^{t-k,k}_{\tilde{s},s} - \sum_{k,s'} x^{t,k}_{s,s'} \Big]. \tag{29}$$

7.3.2 ACTUAL DEMAND

We only know the satisfied demand from existing data sets. Specifically, when a base station becomes empty, the unobserved lost demand is not captured in the data sets.

Previous works in inventory management have represented and verified the random arrival of customer demand following a Poisson process. More importantly, earlier works in bike sharing (Kabra, Belavina, & Girotra, 2015; Shu et al., 2013; George & Xia, 2011) have also represented the random arrival of customers at each station and at each time step using a Poisson distribution and assumed that customers choose their destination station with a certain probability. In a similar vein, we also represent the arrival of customers at a base station in a time step using a Poisson distribution. Since we can only know about the satisfied demand from the data sets, the mean of the Poisson distribution is the average served demand (outgoing flow) in that time step. If $f^{t,k}_{s,s'}$ denotes the average number of bikes booked from station s at time step t and reached station s' at time step $t+k$, then the total outgoing flow from station s at time step t is given by $O^t_s = \sum_{s',k} f^{t,k}_{s,s'}$.

Formally, a demand scenario at station s at time step t (denoted by D^t_s) is generated from a Poisson distribution with a mean of O^t_s (i.e., $D^t_s = \text{Poisson}(O^t_s)$). Finally, the flow from station s to s' at time step t is calculated as the product between outgoing flow from station s and the probability of moving from s to s' at time step t (i.e., $F^{t,k}_{s,s} = D^t_s \cdot \frac{f^{t,k}_{s,s'}}{\sum_{k,s'} f^{t,k}_{s,s'}}$).

7.4 Evaluation Methodology

We employ the following two general steps to evaluate our approach:

1. We compute repositioning and routing solution based on the DRRP tuple that is populated from the training data set.

2. The computed solutions are then evaluated on a simulation using the test data set. That is to say, transitions in the simulation follow the aggregate transition dynamics observed in testing data set where the demand scenarios are generated from Poisson distribution. The evaluation is then aggregated over these generated demand scenarios.

In cases where we do not have sufficient data, we calculate a solution based on the entire data set and we evaluate our solution on various samples from the Poisson distribution with the mean computed from that data set.

13. Note that since we are repositioning bikes, the actual number of bikes at stations will not be the same as the one observed in data.

For the online benchmark approach of Schuijbroek et al. (2017), the next time step solution for moving vehicles and bikes (recommended by the repositioning strategy) is executed using the current positions of bikes and vehicles in the simulation based on the test data set.

For the static method, we employ the simulation to compute the flows of bikes in each time step when no repositioning is done and use that flow information to calculate the expected profit and lost demand. Given the aggregated flow $\mathbf{F}$ and the actual flow $\mathbf{x}$, the revenue is computed as $\sum_{t,k,s,s'} \left[R^{t,k}_{s,s'} \cdot x^{t,k}_{s,s'}\right]$, while the lost demand is computed as $\sum_{t,k,s,s'} \left[F^{t,k}_{s,s'} - x^{t,k}_{s,s'}\right]$.

8. Experimental Results

In this section, we verify the following claims[14]:

1. In terms of scalability, the LDD improves over the MILP and the use of Abstraction on top of the LDD further improves the performance. In terms of solution quality, both the LDD and Abstraction obtain near optimal solutions.

2. Our dynamic approach (LDD + Abstraction of MILP) improves upon the two benchmark approaches (static and online) in terms of lost demand and profit.

3. Our approach remains robust with respect to changes in other input parameters such as the number of vehicles and the unit cost for routing.

8.1 Utility of LDD and Abstraction

To validate the claim that the LDD and Abstraction both improve the original MILP, we provide three sets of results. As the MILP with and without the LDD can only solve small problem instances, we provide these results on the synthetic data sets generated from the *Capital Bikeshare* data set.

Runtime performance: First, we compare the runtime performance of the LDD (SOLVELDD()) with the global MILP (SOLVEDRRP()) in Figure (6a). The X-axis denotes the scale of the problem where we varied the number of stations from 5 to 50. The Y-axis denotes the total time taken to solve the problem in seconds on a logarithmic scale. Except on small instances (e.g., 5-10 stations), the LDD outperforms the global MILP with respect to runtime. More specifically, the global MILP was unable to finish within a cut-off time of 6 hours for any problem with more than 20 stations, while the LDD was able to obtain near optimal solutions on problems with 50 stations in less than an hour.

Duality gap: In the second set of results, we demonstrate the convergence of the LDD to near optimal solutions. The LDD achieves an optimal solution if the duality gap, i.e., the gap between primal and dual solutions, becomes zero. Figure (6b) shows that the duality gap for the instance with 20 stations is only 1%. Figure (6c) and (6d) depict the duality gap for the real-world data set of *Hubway* (with 95 base stations and grouped into 25 abstract stations) and of *Capital Bikeshare* (with 305 base stations and grouped into 50 abstract stations), respectively. For these larger problems we are able to obtain a solution with the duality gap of less than 0.5%.

Effect of abstraction: Finally, we demonstrate the performance of the abstraction method in comparison with the optimal solution of an instance with 30 base stations. We grouped these 30 base stations into 8 abstract stations. Then we run the LDD based optimisation on both the base station and abstraction station problems. Table (12) shows the effect of the abstraction approach on the generated profit and runtime based on five random scenarios of customer demand. With abstraction, while there is only a reduction of less than 0.3% in profit on average from the optimal solution, it gives a significant computational gain.

14. All the linear optimisation models were solved using IBM ILOG CPLEX Optimisation Studio V12.5 incorporated within python code on a 3.2 GHz Intel Core i5 machine.

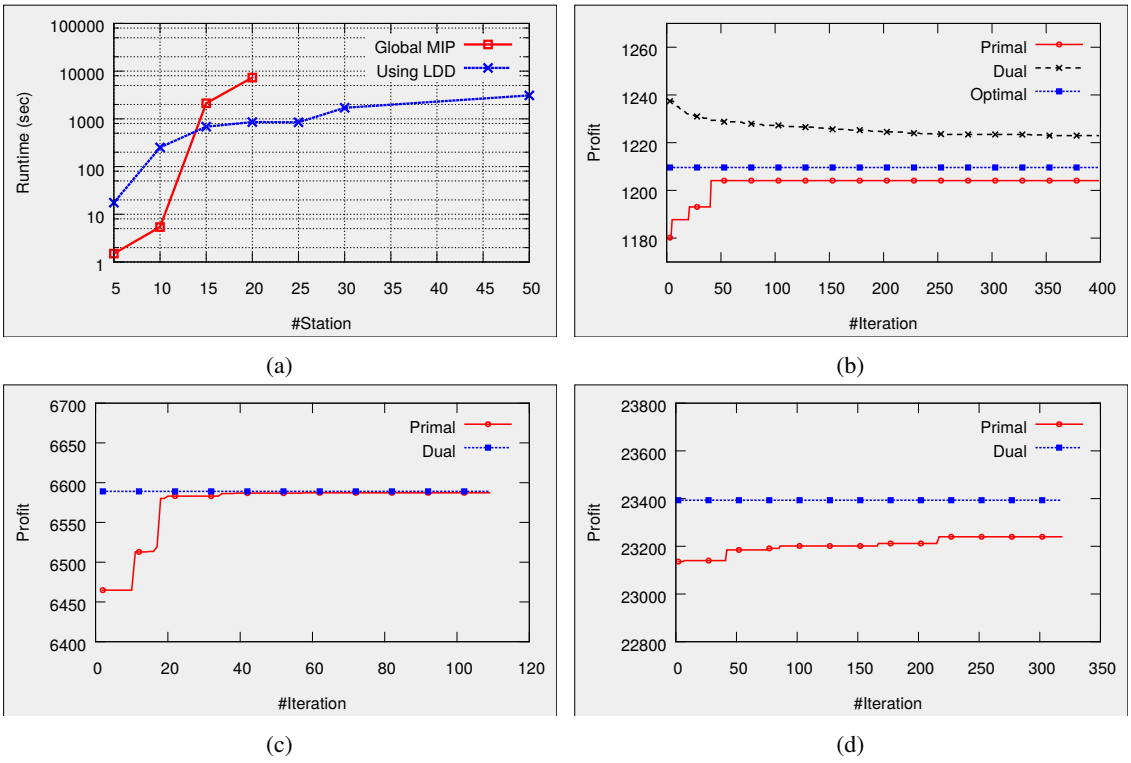


Figure 6: (a) Runtime comparison between the global MILP and LDD, (b) duality gap in the synthetic data set with 20 stations, (c) duality gap in the *Hubway* data set, (d) duality gap in the *Capital Bikeshare* data set.

	With abstraction		Without abstraction		
Instance	**Profit**	**Runtime (sec)**	**Profit**	**Runtime (sec)**	**Profit loss for abstraction**
1	23580	51	23640	3840	0.25%
2	23627	106	23678	3540	0.21%
3	23610	57	23727	3120	0.5%
4	23613	49	23645	3150	0.13%
5	23519	45	23590	3119	0.30%
Average	23590	62	23656	3354	0.27%

Table 12: Effect of abstraction.

The key reason behind this negligible loss of demand when using the abstraction technique is the specific demand patterns observed in the real-world data sets. As shown in the heat maps of Figure (4), the stations that become empty in a particular time period are typically close to each other and hence can be rebalanced within a time step. Since our abstraction is based on geographical proximity, it is ideally suited to handle such situations.

However, the solution quality of our geographical proximity based abstraction mechanism deteriorates if most of the abstract stations become empty at the same time. To demonstrate this situation, we generate artificial demand scenarios where we readjust demand and have high demand for one random station in each abstract station. Table (13) demonstrates the performance of our abstraction approach in comparison with the optimal solution. Even in this small example scenario (with 30 stations) for these artificially crafted demand

instances, we observe a higher reduction (more than 3.2% on average) in the profit due to the abstraction in comparison with the case where we consider the real-world demand (Table 12).

	With abstraction		Without abstraction		
Instance	**Profit**	**Runtime (sec)**	**Profit**	**Runtime (sec)**	**Profit loss for abstraction**
1	11730	21	12092	2918	2.99%
2	11958	46	12314	3201	2.89%
3	11759	24	12114	3050	2.93%
4	11530	36	12060	1641	4.39%
5	11658	31	11997	1467	2.83%
Average	11727	32	12115	2455	3.21%

Table 13: Effect of abstraction on artificially crafted demand.

In this paper we show that our geographical proximity based abstraction mechanism significantly outperforms the existing benchmark approaches due to the specific demand patterns observed in both the real-world data sets of our study. However, our solution approaches are complementary to any abstraction mechanism that can be used to group base stations to reduce the size of the DRRP.

8.2 Comparison against Benchmarks

In this section, we provide the following key comparison results of our approach (dynamic) with the two benchmarks (static and online):

1. Results with respect to profit and lost demand
2. Sensitivity results over different demand scenarios generated from a Poisson distribution
3. Sensitivity results with respect to additional unknown demand
4. Sensitivity results with respect to additional known demand

8.2.1 RESULTS WITH RESPECT TO PROFIT AND LOST DEMAND

We first provide the average results on the *Hubway* and *Capital Bikeshare* data sets for the static, online and our dynamic approach. As indicated earlier, our key performance evaluation metric is the lost demand. However, we also provide the performance comparison with respect to the overall profit to show that we can reduce the lost demand without incurring extra cost to the operators. *Hubway* system consists of 95 active stations and *Capital Bikeshare* system consists of 305 active stations. In our approach, we employ k-means clustering to generate 25 and 50 abstract stations, respectively. Stations within an abstract station are typically within a kilometer of each other for both data sets. For fairness in comparison, we allow a vehicle to visit multiple stations in one time step for the online approach. This is because vehicles are allowed to visit all the stations within an abstract station in one time step in our approach. In fact, we provide a reasonable advantage for the online approach by allowing it to visit 5 stations (anywhere in the city) within one time step[15]. Based on the information obtained from Schuijbroek et al. (2017), we employ 5 vehicles for the experiments on *Capital Bikeshare* data set and 3 vehicles for the experiments on *Hubway* data set. We show the results during the peak period and also for the entire day[16]. See Appendix B for the detailed results on these data sets.

15. This allows for an average distance travelled in one time step with the online heuristic as 17.8 kilometers as opposed to 13.8 kilometers with our approach. Even with this advantage, we demonstrate that our approach performs better.
16. The planning horizon for our approach is 38 time steps (30 minute intervals during the working hours from 5AM-12AM) for the entire day and 14 time steps for the peak period (30 minute intervals during the morning working hours from 5AM-12PM).

	Whole day (5AM-12AM)				Peak period (5AM-12PM)			
	Gain over static repositioning		Gain over online heuristic		Gain over static repositioning		Gain over online heuristic	
	Profit gain	**Lost demand reduction**	**Profit gain**	**Lost demand reduction**	**Profit gain**	**Lost demand reduction**	**Profit gain**	**Lost demand reduction**
Hubway	3.47%	45.80%	3.02%	41.17%	9.16%	46.21%	7.15%	44.75%
Capital Bikeshare	2.14%	22.33%	1.4%	9.9%	4.52%	26.38%	0.96%	5.11%

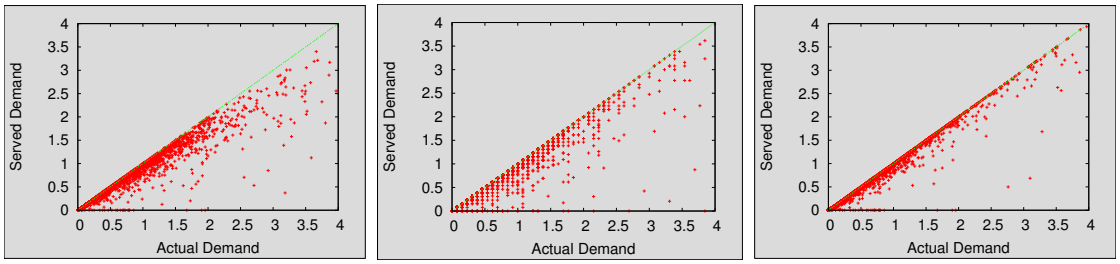
Table 14: Profit and lost demand comparison (*Hubway* and *Capital Bikeshare* data sets).

Table (14) shows the average percentage gain in profit and reduction in lost demand with our approach in comparison to the benchmark approaches on the two real-world data sets. The performance gain in comparison with static repositioning is computed as follows:

$$\text{Profit\_gain} = \frac{\text{Profit using dynamic repositioning} - \text{Profit using static repositioning}}{\text{Profit using static repositioning}}$$

$$\text{Lost\_Demand\_gain} = \frac{\text{Lost demand using static repositioning} - \text{Lost demand using dynamic repositioning}}{\text{Lost demand using static repositioning}}$$

Based on the aggregate results, our approach (LDD + Abstraction) is always able to outperform both the static and online repositioning solutions with respect to both the profit gain and lost demand. Over the entire day, our approach reduces the lost demand in the *Hubway* data set by at least 45.80% and 41.17% in comparison to the static and online approaches, respectively. For the *Capital Bikeshare* data set, we improve by 22.33% and 9.9% in comparison to the static and online approaches, respectively. Similar results (slightly inferior) were obtained when we considered only the peak hour as the planning period.



(a) (b) (c)

Figure 7: Correlation of demand and supply: (a) static repositioning, (b) repositioning using online heuristic, and (c) dynamic repositioning.

Lastly, to visualise the effect of repositioning, we draw the correlation between the actual demand and the served demand over the entire planning horizon. Figure (7) shows the correlation between the actual demand and the demand served by following the three approaches. Each point in the graphs corresponds to the values of an actual demand and its corresponding served demand for all time steps and in all stations in the *Hubway* data set. Therefore, it is better if more points are closer to the identity line ($x = y$). As can be noted, our approach has significantly more points closer to the identity line than the two benchmarks and therefore, is able to better match the supply of bikes with the demand for bikes.

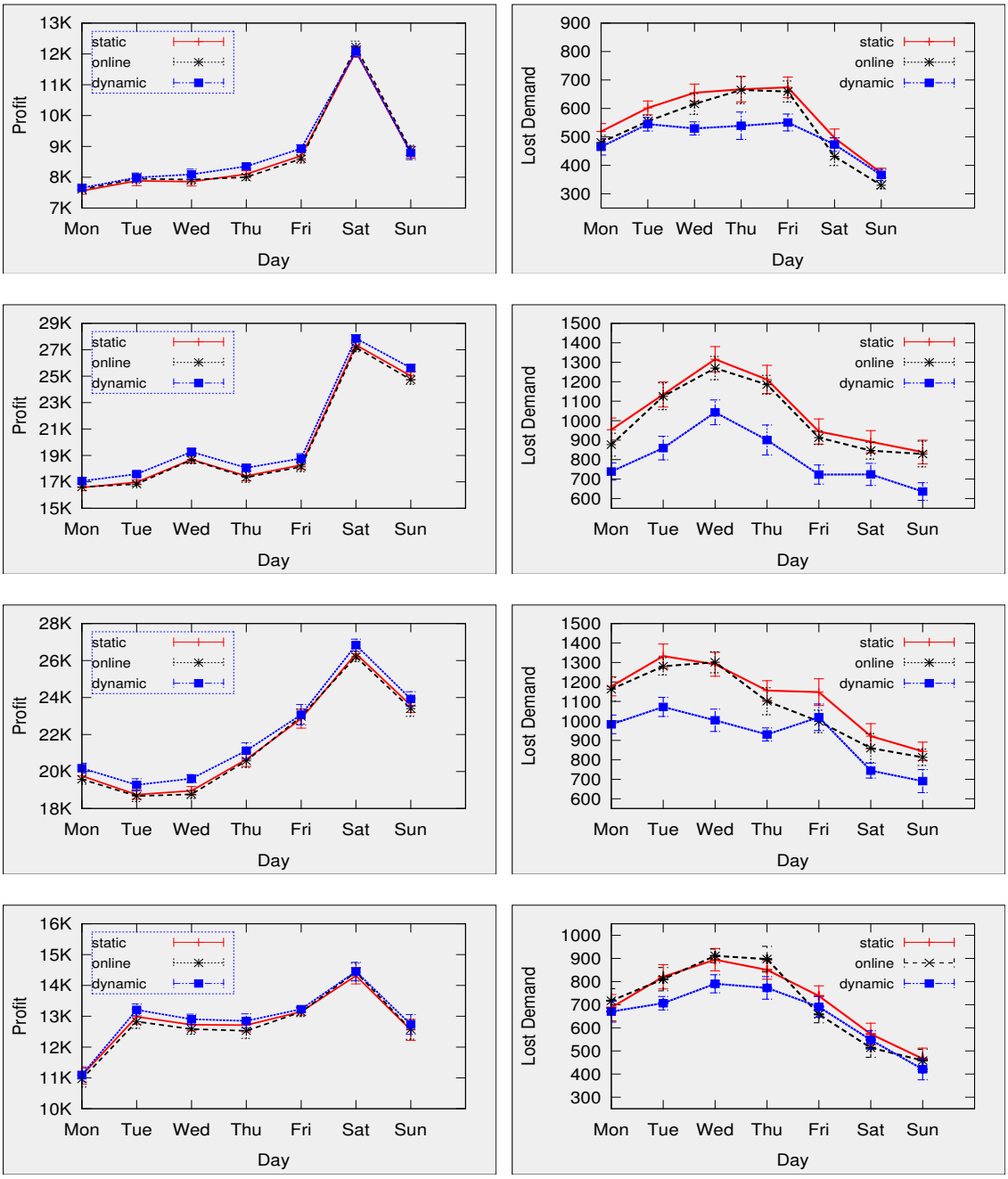


Figure 8: Sensitivity analysis (data set: *Capital Bikeshare*, 4 quarters of 2013): (a) profit comparison, and (b) lost demand comparison.

8.2.2 SENSITIVITY RESULTS OVER DIFFERENT DEMAND SCENARIOS GENERATED FROM POISSON DISTRIBUTION

We now demonstrate the sensitivity of our approach with respect to different demand scenarios. We created ten demand scenarios for each of the weekdays from the underlying Poisson distribution with satisfied

demand as the mean (refer to Section 7.3.2 for details). For each demand scenario, we calculate the profit and lost demand for the benchmark approaches and our approach. Figure (8) shows the mean along with error bars for profit and lost demand for the four quarters of 2013 of the *Capital Bikeshare* data set. The key observations are as follows:

- Our approach (dynamic) is able to provide significantly better results with respect to reduction in lost demand than the two benchmarks on almost all the cases.
- The only cases where the online approach performs better than our approach with respect to lost demand is in quarters 1 and 4 (where the demand was significantly lower than in quarters 2 and 3) and specifically on weekends. On weekends, there is higher variance and inconsistency in demand, so our solution computed using the average demand is unable to adapt as well as the online approach.
- In terms of profit, while the difference is small, our approach is always better than the two benchmarks.

Therefore, even considering the variance, our approach provides a significant reduction in lost demand compared to the two benchmarks.

8.2.3 SENSITIVITY RESULTS WITH RESPECT TO ADDITIONAL UNKNOWN DEMAND

For each day of the week, we evaluate our solution when demand scenarios are modified to include artificial demand. We generate our solution by considering the mean of the historical trip data that does not consider the additional artificial demand. This artificial demand is added to a station in a time step, if that station was observed to be empty at that time interval in the data. Specifically, if a station s is observed to be empty at time step t on one day, then $\alpha\%$ of the mean served demand, F_s^t is added to that station. The destination station and booking period for the newly generated demand are chosen based on the distribution observed in the historical data.

	Gain over static repositioning		Gain over online heuristic	
$\alpha\%$	**Profit gain**	**Lost demand reduction**	**Profit gain**	**Lost demand reduction**
10	1.54%	23.86%	0.95%	15.73%
20	1.55%	23.78%	0.95%	15.06%
30	1.54%	23.45%	0.82%	13.97%
40	1.54%	22.78%	0.75%	12.77%
50	1.53%	22.34%	0.73%	11.89%
60	1.42%	20.81%	0.55%	9.83%
70	1.32%	19.97%	0.46%	8.73%
80	1.27%	18.76%	0.42%	7.78%
90	1.3%	18.37%	0.37%	6.75%
100	1.26%	16.54%	0.28%	4.8%

Table 15: Sensitivity analysis with respect to unknown increase in mean demand (data set: *Capital Bikeshare*).

In Table (15), we provide the comparison results between our approach, static repositioning and online heuristic for one of the weekdays where we vary α from 10% to 100%. The most important result is that even at 100% increase in demand at the empty stations, our dynamic approach performs better by at least 5% in terms of reducing lost demand. Furthermore, the drop in performance as unknown demand increases is gradual.

8.2.4 SENSITIVITY RESULTS WITH RESPECT TO ADDITIONAL KNOWN DEMAND

Predicting unobserved lost demand is a challenging issue in many real-world planning problems including retail planning. Many heuristic methods are mentioned in the literature (Kök & Fisher, 2007; Musalem, Olivares, Bradlow, Terwiesch, & Corsten, 2010; Vulcano, Van Ryzin, & Ratliff, 2012) to predict the unobserved lost demand. Our repositioning approach is not dependent on the method employed to predict mean demand. So, we can always complement our approach with the best approach from the literature for predicting the mean demand.

We could also apply simple heuristics to learn demand values over time. We can consider small increments in mean demand for those stations and time steps when they become empty. For example, if station X typically becomes empty (say observed over a month) at a time step, we then consider the mean demand for station X as 102% of the realised demand at that time step. Over time, if we still observe that the station becomes empty at that time step, then we consider a mean demand that is further 2% over the realised demand. Such an approach over time will converge to the actual demand.

Furthermore, some bike sharing systems (e.g., *Bixi* in Montreal) are considering an operational enhancement that will further alleviate the problem of identifying the actual demand. In this enhancement, if customers encounter an empty or congested station, there is a provision for them to enter this information in the system that is installed at each of the base stations (assuming there is an incentive for riders to provide their information). With this minor operational enhancement, the accuracy of actual demand will increase significantly and most importantly for this paper, our approach will benefit from higher accuracy on predicting the exact demand values.

	Gain over static repositioning		Gain over online heuristic	
$\alpha\%$	**Profit gain**	**Lost demand reduction**	**Profit gain**	**Lost demand reduction**
10	1.66%	29.69%	1.74%	27.44%
20	1.63%	30.09%	1.78%	28.94%
30	1.67%	28.24%	1.48%	22.75%
40	2.20%	34.26%	1.40%	23.33%
50	1.56%	24.57%	0.96%	16.38%
60	2.03%	32.32%	1.15%	22.03%
70	2.28%	32.61%	1.64%	25.41%
80	1.63%	25.61%	0.82%	16.64%
90	1.92%	27.64%	0.26%	11.52%
100	2.02%	22.78%	0.28%	7.30%

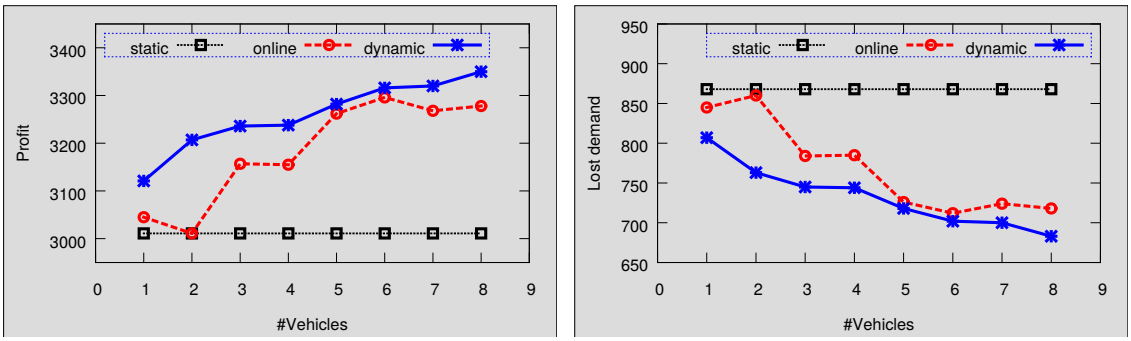
Table 16: Sensitivity analysis with respect to known increase in mean demand (data set: *Capital Bikeshare*).

In order to demonstrate generality, we now provide a detailed comparison between our solution, static repositioning and online heuristic by assuming that the extra demand is known a priori using one of the methods provided in the previous paragraphs. We employ the same mechanism to introduce extra artificial demand using α parameter as described in Section 8.2.3. However, since the demand is known beforehand, it is taken into consideration in our approach as well as in the online heuristic to compute repositioning and routing strategies. In Table (16), we provide the comparison results with respect to profit and lost demand while α is varied from 10% to 100%. As clearly shown in Table (16), our approach provides better results for all values of α in comparison to the static and online approaches.

8.3 Performance Comparisons with Changes in Parameters

The performance of the repositioning solution is reliant upon input parameters such as the number of vehicles, unit cost for routing and duration of each time step. In this section, we describe the effect of those input parameters on key performance metrics such as profit earned by the operator and the lost demand.

Effect of the number of vehicles: To understand the effect of the number of vehicles we compare the performance of the three approaches (static, online and dynamic) with different numbers of carrier vehicles. Figure (9) shows the analysis of profit and lost demand on a synthetic data set with 20 stations. Figure (9a) shows that the profit obtained by using our approach increases monotonically as we increase the number of vehicles. Although the profit gain of our approach in comparison to the online approach fluctuates due to the myopic nature of the online heuristic, the gain is always positive. Figure (9b) shows a similar pattern in the performance with respect to lost demand. Lost demand reduces monotonically for our approach as the number of vehicles is increased and the gain in reducing lost demand for our approach over both the static and the online approach is always positive.



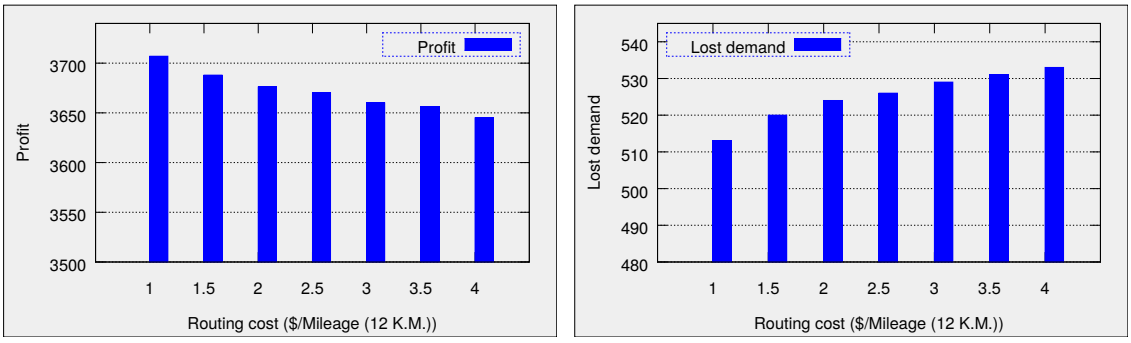
(a) (b)

Figure 9: Effect of the number of vehicles on (a) profit, and (b) lost demand.

Effect of routing cost: Routing cost, P is an important parameter in our optimisation model. While some bike sharing operators outsource the repositioning tasks to other agencies that charge a certain amount for moving individual bikes, most BSS operators use their own vehicles and the cost of repositioning is equivalent to the routing cost for the vehicles. In this section, we provide a sensitivity analysis with respect to the fuel price (i.e., dollar cost per 12 kilometers of routing which is the average mileage of vehicles as shown by Fishman et al., 2014) on a synthetic data set with 20 stations. Figure (10) plots the profit and lost demand when we vary the unit fuel cost from 1 dollar to 4 dollars on the X-axis. As expected, Figure (10a) shows that the profit earned by the operator decreases as we increase the unit cost of routing. Furthermore, Figure (10b) depicts that the lost demand increases by a small amount if the unit cost for routing is increased.

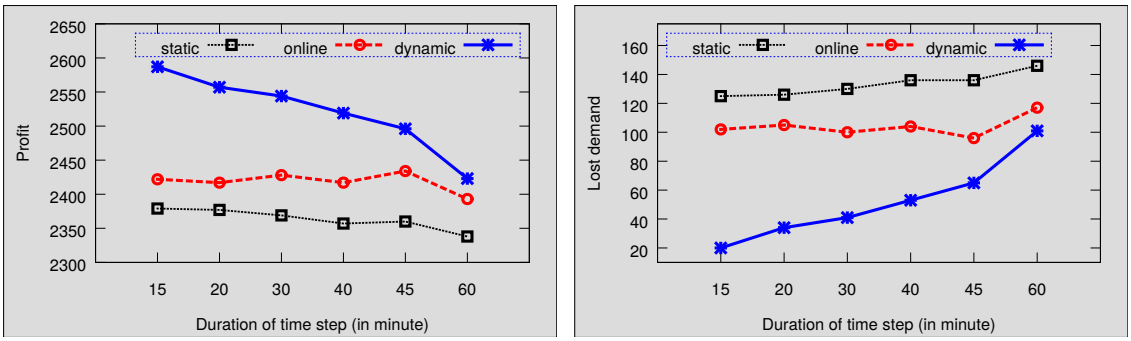
Effect of duration of time step: We now provide an analysis on the profit and lost demand, when the duration of time step is varied. Figure (11) plots the performance metrics when we vary the duration of time step from 15 minutes to one hour on the X-axis. Figure (11a) shows that the profit for the operator reduces monotonically as we increase the duration of time step. Increasing the duration of time step entails vehicles can visit and rebalance a fewer number of base stations and therefore, produces lower profit for the operator. Figure (11b) shows that the lost demand increases significantly if we increase the duration of time step. Most importantly, performance gain of our approach over the static repositioning and online heuristic increases monotonically as we reduce the duration of time step. This can be attributed to our dynamic approach making better use of the extra repositioning opportunities (due to shorter duration) and promptly react to future demand changes.

On the other hand, reducing the duration of time step notably increases the runtime. For example, the runtime of the problems with 15 minutes of time step is approximately 2 hours while the problems with



(a) (b)

Figure 10: Effect of routing cost on (a) profit, and (b) lost demand.



(a) (b)

Figure 11: Effect of the duration of time step on (a) profit, and (b) lost demand.

30 minutes of time step are solved within 30 minutes. So, there is a clear trade-off between utility and runtime in deciding the right duration of time step. Although the performance in terms of profit and lost demand decreases by a small amount for 30 minutes of time step (over 15 minutes of time step), it provides a significant computational gain and is particularly helpful when solving large problems. Therefore, we choose 30 minute as the default setting for the duration of time step.

9. Model Extensions and Supplementary Analysis

In this section, we discuss ways of relaxing some of the assumptions made in the generic formulation of Table (4). We further provide a discussion on potential extensions.

9.1 Accounting for Physical Limitations in Vehicle Movement

In the MILP of Table (4) we assume that a vehicle can travel between any pair of stations within one time step without considering their relative distance. For the two data sets we considered, the average distance between any two stations is approximately 2 miles for *Hubway* and 5 miles for *Capital Bikeshare*[17], so our assumption of being able to travel within 30 minutes is reasonable and conservative (that accounts for the

17. The maximum distance between any two stations was 18 miles and 90th percentile of the distances between any two stations are within 10 miles for both the data sets. So, even in the worst case any two stations could be covered within 30 minutes. Furthermore, our solutions (rather any good solution) would not recommend a carrier vehicle to travel the maximum distance to shift the bikes.

time to load and unload bikes). However, in other settings, it may not be the case and there might be multiple time steps needed to cover certain stations.

In this segment, we provide a minor update to the previous formulation which is able to account for physical limitations in vehicle movement. We introduce a new set, $\mathcal{B}_s^{\hat{t}}$ to capture the physical reachability of stations in a certain number of time steps. Specifically, $\mathcal{B}_s^{\hat{t}}$ denotes the set of stations which can be reached within $\hat{t}$ time steps from station s. The modified optimisation model is provided in Table (17). Amongst the constraints that consider vehicle movements in the original formulation of Table (4), the ones that must be modified due to physical limitations of vehicle movements are the vehicle flow conservation constraints (5). Essentially, the change in updated constraints (30) reflects the fact that we only need to consider vehicle transitions from stations that are reachable in a given number of time steps (and not others).

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

s.t. *Constraints* (2)- (4) *hold*

$$\sum_{k \in \mathcal{S}} z_{s,k,v}^{t} - \sum_{\hat{t}} \sum_{k \in \mathcal{B}_s^{\hat{t}}} z_{k,s,v}^{t-\hat{t}} = \sigma_{v,s}^{t}, \qquad \forall s,t,v \tag{30}$$

Constraints (6)- (9) *hold*

Table 17: RESTRICTEDDRRP()

Accounting for physical limitations potentially entails finer division of time steps and hence the number of time steps increases. However, the number of transitions between stations at any one time step is reduced. Therefore, as we show below, accounting for physical limitations does not have a significant effect on the scalability of our approach. Since the inherent assumption of reachability is different, we primarily compare the runtimes in Table (18) to verify the claim on scalability.

	Runtime with physical limitations	Runtime without physical limitations
Mon	5120	4880
Tue	5183	3951
Wed	4136	4966
Thu	5245	4980
Fri	5127	3992
Average	4962	4554

Table 18: Effect of physical limitations in vehicle movements on runtime (in seconds).

We consider the *Hubway* data set to run the scalability experiments. When considering physical limitations, we assume that all the stations can be reached within 3 time steps at the maximum and the number of stations reachable in one time step from any given station is decided based on their relative distance. When considering no physical limitations, we assume that all stations are reachable from any other station in one time step. We observe that the approach considering physical limitations usually takes longer, however, the difference is not significant and consistent. On average the approach considering physical limitations takes 10% more time to find a solution.

Another simple mechanism that can be adopted to deal with physical limitations without making changes to our approach is based on clustering of stations. We can cluster stations that can all be reached in one time step into one zone and assign a set of vehicles to that zone. This way, we can apply our method directly to each zone.

A minor modification to the MILP of Table (17) can be used to represent vehicles taking different times to move between stations at different times of the day. For instance, during peak hours, a vehicle might take longer to move between stations in the city. To model such scenario, we need to replace the set $B_s^{\hat{t}}$ with set $B_{t,s}^{\hat{t}}$ which contains all the stations that can be reached within $\hat{t}$ time steps if a vehicle starts from station s at time step t. The only modification required in the optimisation model is in the constraints (30). This is to compute the inflow of vehicles at station s at time step t by considering all the stations from where a vehicle should take $\hat{t}$ time steps to reach station s if it has started its journey at time step $t - \hat{t}$ (i.e., all the elements of the set $B_{t-\hat{t},s}^{\hat{t}}$).

9.2 Different Time Scales for Vehicle and Bike Movements

$$\min_{\mathbf{y}^+,\mathbf{y}^-,\mathbf{z}} - \sum_{t,k,s,s'} R_{s,s'}^{t,k} \cdot x_{s,s'}^{t,k} + \sum_{\hat{t},v,s,s'} P_{s,s'} \cdot z_{s,s',v}^{\hat{t}} \tag{31}$$

$$\textbf{s.t.} \quad d_s^{\#,t} + \sum_{k,\hat{s}} x_{\hat{s},s}^{t-k,k} - \sum_{k,s'} x_{s,s'}^{t,k} + \sum_{\hat{t}=m.(t)}^{m.(t+1)} \sum_{v} (y_{s,v}^{-,\hat{t}} - y_{s,v}^{+,\hat{t}}) = d_s^{\#,t+1}, \qquad \forall t, s \tag{32}$$

$$x_{s,s'}^{t,k} \leq d_s^{\#,t} \cdot \frac{F_{s,s'}^{t,k}}{\sum_{k,\hat{s}} F_{s,\hat{s}}^{t,k}}, \qquad \forall t, k, s, s' \tag{33}$$

$$d_v^{*,\hat{t}} + \sum_{s \in S} [(y_{s,v}^{+,\hat{t}} - y_{s,v}^{-,\hat{t}})] = d_v^{*,\hat{t}+1}, \qquad \forall \hat{t}, v \tag{34}$$

$$\sum_{k \in \mathcal{S}} z_{s,k,v}^{\hat{t}} - \sum_{k \in \mathcal{S}} z_{k,s,v}^{\hat{t}-1} = \sigma_{v,s}^{\hat{t}}, \qquad \forall \hat{t}, s, v \tag{35}$$

$$\sum_{j \in \mathcal{S}, v \in \mathcal{V}} z_{s,j,v}^{\hat{t}} \leq 1, \qquad \forall \hat{t}, s \tag{36}$$

$$y_{s,v}^{+,\hat{t}} + y_{s,v}^{-,\hat{t}} \leq C_v^* \cdot \sum_i z_{s,i,v}^{\hat{t}}, \qquad \forall \hat{t}, s, v \tag{37}$$

$$0 \leq x_{s,s'}^{t,k} \leq F_{s,s'}^{t,k}, 0 \leq d_s^{\#,t} \leq C_s^{\#}, 0 \leq y_{s,v}^{+,\hat{t}}, y_{s,v}^{-,\hat{t}} \leq C_v^*, 0 \leq d_v^{*,\hat{t}} \leq C_v^* \tag{38}$$

$$z_{i,j,v}^{\hat{t}} \in \{0, 1\} \tag{39}$$

Table 19: SOLVEDRRPDIFFTIMESCALES()

Our original formulation in Table (4) assumes that the time scale for customer movements of bikes and vehicle movements is the same. In practice, a vehicle can reposition bikes from multiple stations in each time step. Therefore, we now provide a general formulation in Table (19), where the bikes and vehicles are operating on different time scales. Except for the two different time scales where a vehicle is assumed to travel m stations at each time step (i.e., $t = m \cdot \hat{t}$), the structure of the formulation is similar to the one in Table (4). Therefore, the enhancements provided with respect to decomposition and abstraction are applicable in similar ways. We further note that the opposite case where the time scale of the movements of bikes is smaller than the one for the vehicles can basically be solved by our original formulation by aggregating the customer incoming and outgoing flows of bikes over the vehicle time scale. This is due to the fact that no rebalancing can be made during that interval.

9.3 Approximate Customer Flow Dynamics in Solution Computation

Since we maximise profit, we can identify boundary cases where bikes are not rented at certain time steps even though demand is present. Such cases can arise in our solution to save bikes for a later time step when it

is possible to get higher profit. However, they do not appear in our evaluation because we have a data-driven approach where we evaluate on a test data set (that is different from training data set). Therefore, accounting for real dynamics in training data set is not always necessary. Additionally, accounting for exact dynamics increases the computational complexity of the solution approach significantly. In our experimental results, we show that 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 (refer to constraints (40)) that ensure total outflow of bikes from station s at time step t should be equal to the minimum of total arrival demand and the number of bikes present at source station. But constraints (40) are quadratic in nature and our MILP becomes a higher order conic program.

$$\sum_{k,s'} x_{s,s'}^{t,k} = \min(d_s^{\#,t}, \sum_{k,s'} F_{s,s'}^{t,k}), \quad \forall t, s \tag{40}$$

Apart from being quadratic, as mentioned by Shu et al. (2013), constraints (40) can only be the sufficient condition to handle the real dynamic of BSSs if stations have unlimited bike docking capacity. Because of these difficulties, we focus on representing bike flow dynamics approximately in our optimisation model.

9.4 Offline Solution, Online Execution

Note that when executing the repositioning solution computed offline, the operator may find that the state of the system is different from what it was assumed to be, so the plan may not be feasible. Furthermore, this infeasibility reflects on other stations as well. For instance, the number of bikes left in the vehicle is smaller or larger than planned. We employ online modifications to deal with such situations at execution time: (1) the number of bike pickup at any time step is set as the minimum value between the number of empty slots in vehicle, the number of bikes present in the station and the number of planned pickup, and (2) the number of drop-offs at any time step is set as the minimum value between the number of bikes in the vehicle, the number of empty docks in the station and the number of planned drop-off at that time step.

As demonstrated in the sensitivity analysis results, even with such modifications to solution at execution time, our solutions are still able to provide non-trivial improvements in terms of lost demand and profit over the benchmark approaches. This is because the pattern of demand is consistent when compared over similar days (i.e., Monday pattern with another Mondays) and does not have a huge variance except on weekends. Due to this reason, there is no cascade effect when we make local changes to our solution.

9.5 Objective

The objective employed in SOLVEDRRP() represents a trade-off between two objectives, namely, maximising serviced demand and minimising routing cost. We combine these two objectives based on their dollar value. Specifically, we use a revenue model derived from the model employed by the real system to calculate the dollar value of serviced demand and a cost model derived from prevailing fuel costs to calculate the dollar value of routing cost. It should be noted that this is just one way of combining the two components and there can be other ways of combining the two components. In this paper, we focus on this one combination of the two objectives. As shown in the experimental results, this way of combining the two components significantly improves the serviced demand and also the combined profit of the two components. In the future, we plan on exploring the Pareto-front of these two objectives.

9.6 Labor Cost

There are two levels of decision making involved in long-term and large facility investments such as bike sharing systems:

- Strategic level: capital and labor decisions, i.e., how many vehicles, bikes, etc. to buy and how many permanent employees to hire?

- Operational level: day-to-day decisions associated with routing of vehicles and repositioning of bikes, i.e., how many bikes to reposition, from where and how?

Strategic level decisions consider long-term profits and typically do not change on a daily basis. Operational level decisions change on a daily basis and are the key focus of this paper. We provide a quick example to demonstrate that long term reasoning (and not day-to-day reasoning) with respect to labor costs is a better option. The US Department of Labor[18] provides hourly and yearly salaries for drivers operating light trucks or other delivery services in US. If we consider that a driver is hired for 6 hours (the time required to reposition bikes at the end of the day), the median cost would be 84. However, if the operator hired a driver for a year, the median salary for one day is just 80 (=29170/365). A similar result based on real statistics is available for capital costs (e.g., vehicles). This example entails that dynamic repositioning throughout the day or repositioning at the end of day would have similar labor costs. Also, since labor/capital costs would be constant in the optimisation model of Table (4), they would not alter the results corresponding to lost demand. We also have a buffer on the fuel cost to account for any other costs pertaining to day-to-day operations.

9.7 Operational Enhancement to Our Abstraction Approach

Recent bike sharing systems (e.g., *Citibike* in New York City) have introduced the concept of bike-trailers (O'Mahony & Shmoys, 2015) that can reposition a small number of bikes to nearby stations. This operational enhancement can significantly improve the performance of our geographical proximity based abstraction scheme. As the bike-trailers are only used to match the need of nearby producer and consumer stations, they can be used effectively to balance all the base stations within an abstract station. Essentially, larger vehicles are used to rebalance the system at the level of abstract stations while bike-trailers can be used to rebalance within each abstract station.

10. Conclusion

We consider the problem of dynamically repositioning bikes to improve their availability and to reduce the usage of private vehicles. The general insight that we introduce in this paper is that, while performing repositioning, it is useful to consider demand surges and dips during the day. To that end, we use a mixed integer linear programming approach that employs Lagrangian dual decomposition and abstraction mechanisms to provide: (1) a near optimal solution for the dynamic repositioning of idle bikes in conjunction with the routing solution for vehicles during the day, and (2) a scalable solution for the real-world large scale bike sharing systems. The empirical results on multiple real and synthetic data sets show that our dynamic repositioning approach is not only able to achieve the original goal of reducing lost demand, but is also able to improve profit for the bike sharing system. In future, this work can be extended with a robust optimisation technique which can account for all the realisations 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). The authors thank the anonymous referees, the editor, as well as the HEC workshop participants for their valuable comments and suggestions.

Appendix A. Online Approach Based on the Model of Schuijbroek et al. (2017)

In this section, we provide the details of the model used by Schuijbroek et al. (2017) and also how we have adapted the model to solve the problem in our study. The primary goal of Schuijbroek et al. (2017) is to

18. http://www.bls.gov/oes/current/oes533033.htm provides the information of labor cost in US.

minimise the operational cost for the routing of vehicles such that the whole system can be balanced. The key aspects of their approach are as follows:

- Customer movements at the time of rebalancing operation are negligible.
- Vehicles can visit all the stations to rebalance the whole system within the rebalancing period.
- Each vehicle is assigned to a group of base stations. The entire system is divided into a set of cluster (the number of clusters is equals to the number of vehicles), each of which is allocated with a vehicle. Thus, a vehicle is only responsible for the repositioning of bikes within a particular cluster and it can visit all the base stations of that cluster within the rebalancing period.

With minor changes we are able to adapt their approach to the dynamic repositioning context. Intuitively, we make changes corresponding to the first two points above and leave the rest of the approach as it is:

- Because of the dynamism and significant customer movements during the rebalancing period, the demand model assumed by Schuijbroek et al. (2017) is not valid in our context. Because of the different demand model assumptions, their approach to compute the inventory level is not applicable for solving the DRRP. However, we figure out the best inventory levels (number of required bikes at stations) through experimentation. More specific details will be mentioned later.
- Since the assumption of visiting all stations to rebalance in one time step is not reasonable, we set a maximum number of stations that can be rebalanced in one time step. Specifically, we set it to 5 stations (any 5 stations can be visited in one time step) and this corresponds to an average distance travelled by a vehicle in one time step to be approximately 3.4 kilometers (which is slightly above the distance travelled with our approach which is around 2.6 kilometers).

$$\min \quad \hat{H} \tag{41}$$

$$\textbf{s.t.} \quad \sum_{v \in \mathcal{V}} z_{i,v} = 1, \qquad \forall i \in \{\mathcal{S} \setminus S^{+}\} \tag{42}$$

$$\sum_{v \in \mathcal{V}} z_{i,v} \leq 1, \qquad \forall i \in S^{+} \tag{43}$$

$$q_v^0 + \sum_{i \in S} s_i^0 z_{i,v} \geq \sum_{i \in S} s_i^{min} z_{i,v}, \qquad \forall v \in \mathcal{V} \tag{44}$$

$$-(C_v^* - q_v^0) + \sum_{i \in S} s_i^0 z_{i,v} \leq \sum_{i \in S} s_i^{max} z_{i,v}, \qquad \forall v \in \mathcal{V} \tag{45}$$

$$\hat{h}_v \geq \sum_{j \in \mathcal{S}} d_{i,j}(z_{i,v} + z_{j,v} - 1), \qquad \forall s \in \mathcal{S}, v \in \mathcal{V} \tag{46}$$

$$\hat{h}_v \geq \hat{H}, \qquad \forall v \in \mathcal{V} \tag{47}$$

$$z_{i,v} \in \{0,1\}, \hat{h}_v \geq 0, \hat{H} \geq 0 \tag{48}$$

Table 20: MAXSPS-BASED-CLUSTERING($\mathbf{s}^0, drrp$)

We now describe the details of the online heuristic approach. To generate the cluster of stations for each vehicle, a maximum spanning star (MAXSPS) based approximation technique is employed. The idea is to minimise the maximum expected routing cost $\hat{H}$ (delineated in Expression 41) such that the service level requirement for each station can be satisfied. An optimisation problem (described in Table 20) is solved to discover the cluster of stations for each vehicle. Let, $z_{i,v}$ be a binary decision variable that is set to 1 if station $i \in \mathcal{S}$ is assigned to the cluster of vehicle $v \in \mathcal{V}$. Let, S^{+} denotes the set of self-sufficient stations

and $\hat{h}_v$ denotes the routing cost for vehicle v. Constraints (42)-(43) ensure that insufficient stations ($\mathcal{S} \setminus S^+$) must be visited by vehicles while sufficient stations can be visited if required. Constraints (44)-(45) ensure that each cluster contains enough bikes such that service level requirement can be satisfied for each station. Constraints (46) estimate the lower bound on the routing cost for the resulting assignment and constraints (47) enforce that the objective value of makespan is equivalent to $\max_v \hat{h}_v$.

$$\min H + w\big(\sum_i \delta_i^+ + \sum_i \delta_i^-\big) \tag{49}$$

$$\textbf{s.t.} \quad s_i^0 + \sum_{t,v}(y_{i,v}^{-,t} - y_{i,v}^{+,t}) \geq s_i^{min} - \delta_i^+, \qquad \forall i \tag{50}$$

$$s_i^0 + \sum_{t,v}(y_{i,v}^{-,t} - y_{i,v}^{+,t}) \leq s_i^{max} + \delta_i^-, \qquad \forall i \tag{51}$$

$$\sum_j z_{i,j,v}^t - \sum_j z_{j,i,v}^{t-1} = \sigma_{v,i}^t, \qquad \forall i,t,v \tag{52}$$

$$\sum_{t,j \notin G_v} z_{i,j,v}^t = 0, \qquad \forall i,v \tag{53}$$

$$y_{i,v}^{+,t} + y_{i,v}^{-,t} \leq C_v^* \cdot \sum_j z_{i,j,v}^t, \qquad \forall i,t,v \tag{54}$$

$$d_v^{*,t} + \sum_i (y_{i,v}^{+,t} - y_{i,v}^{-,t}) = d_v^{*,t+1}, \qquad \forall t,v \tag{55}$$

$$H \geq \sum_{i,j,t} P_{i,j} \cdot z_{i,j,v}^t, \qquad \forall v \tag{56}$$

$$s_i^0 + \sum_{t,v}(y_{i,v}^{-,t} - y_{i,v}^{+,t}) \geq 0, \qquad \forall i \tag{57}$$

$$s_i^0 + \sum_{t,v}(y_{i,v}^{-,t} - y_{i,v}^{+,t}) \leq C_i^{\#}, \qquad \forall i \tag{58}$$

$$0 \leq y_{i,v}^{+,t}, y_{i,v}^{-,t} \leq C_v^*, 0 \leq d_v^{*,t} \leq C_v^*, H, \delta_i^+, \delta_i^- \geq 0, z_{i,j,v}^t \in \{0,1\} \tag{59}$$

Table 21: SOLVEONLINE($\mathbf{s}^0, drrp$)

Table (21) provides the MILP formulation for the online heuristic approach that is used in each time step to generate a repositioning solution. As a vehicle can visit multiple stations within one time step, the time indicators are used in SOLVEONLINE() to represent the sequence of moves of all the vehicles. The objective function (49) is to minimise the maximum routing cost for all the vehicles. As a vehicle cannot visit all the stations within the rebalancing period, we add additional slack variables δ^+, δ^- in the objective to ensure that maximum number of stations are balanced. w represents the unit penalty for deviating from the minimum and maximum number of bikes required at each station and we set it to 1 in our experiments. As the goal of the MILP is to minimise the routing cost as well as the expected lost demand, the approach is comparable to our approach. s_i^0 is the initial distribution of bikes at station i in that time step. s_i^{min}, s_i^{max} represent the lower and upper bounds, respectively, on the number of bikes required at station i. Let, F_i^t is the expected demand at station i in the time step t, then s_i^{min}, s_i^{max} are determined as $(1-\epsilon)F_i^t$ and $(1+\epsilon)F_i^t$, respectively, where ϵ is the tolerance level[19]. A vehicle v is allocated to the cluster G_v.

Constraints (50) ensure that each station has at least the minimum required bikes after the repositioning. As all the stations cannot be balanced within one time step, the slack variable δ^+ is added in these constraints to avoid the infeasibility of the MILP. Constraints (51) ensure that the number of bikes at each station does not exceed the maximum required number of bikes after the repositioning and the slack variable δ^- is added to

19. We take the value of ϵ as 10% in the experiment.

	Whole day (5AM-12AM)				Peak period (5AM-12PM)			
	Gain over static repositioning		Gain over online heuristic		Gain over static repositioning		Gain over online heuristic	
	Profit gain	**Lost demand reduction**	**Profit gain**	**Lost demand reduction**	**Profit gain**	**Lost demand reduction**	**Profit gain**	**Lost demand reduction**
Mon	2.47%	24.53%	2.41%	22.86%	8.08%	43.56%	7.15%	37.41%
Tue	3.62%	35.15%	4.32%	36.87%	13.01%	55.79%	11.17%	52.27%
Wed	3.17%	30.13%	3.20%	29.22%	12.30%	53.76%	9.43%	48.05%
Thu	4.03%	36.93%	3.92%	35.89%	13.32%	52.56%	9.26%	45.16%
Fri	5.63%	50.00%	5.08%	47.06%	16.15%	67.78%	12.22%	61.33%
Sat	2.20%	69.89%	1.18%	58.21%	0.70%	25.00%	0.63%	35.71%
Sun	3.15%	74.00%	1.00%	58.06%	0.53%	25.00%	0.21%	33.33%
Mean	3.47%	45.80%	3.02%	41.17%	9.16%	46.21%	7.15%	44.75%

Table 22: Profit and lost demand comparison (data set: *Hubway*, 3rd quarter of 2012).

these constraints to avoid the above mentioned difficulties. Constraints (52) enforce that the flows of vehicles in and out of stations are preserved. Constraints (53) ensure that vehicle v can only visit stations within the cluster G_v. Constraints (54) ensure that the flows of bikes in and out of vehicles are preserved at the time of repositioning. Constraints (55) enforce that a vehicle can only pick up or drop off bikes at a station if it is present at that station. Constraints (56) ensure that the variable H in the objective is greater than the routing cost for each vehicle or alternatively assure that we minimise the maximum routing cost of the vehicles. Constraints (57)-(58) ensure that the station capacity is not exceeded while repositioning the bikes. Lastly, constraints (59) enforce that the vehicle capacity is not exceeded when repositioning bikes.

In each time step, given the distribution of bikes, we find the repositioning solution by solving the MILP provided in Table (21). After the repositioning, we update the number of bikes in each station and simulate the flows of bikes according to customer demand. Once we determine the flows of bikes, we can compute the distribution of bikes in each station for the next time step and this information is used to execute the MILP of Table (21) for the next time step. The process iterates until we reach the last time step.

Appendix B. Experimental Results

In this section, we present the detailed experimental results corresponding to the average results provided in Section 8.2.1. Specifically, we provide quarterly results of the percentage gain in profit and the percentage reduction in lost demand in comparison with the two benchmarks (static repositioning and online heuristic approach) for the *Hubway* and *Capital Bikeshare* data sets.

We begin with the results on the real-world data set of *Hubway*. *Hubway* BSS comprises with 95 base stations and we group them into 25 abstract stations. We employ 3 vehicles (courtesy: Schuijbroek et al., 2017) for this experiment. We only have the proper trip history data for third quarter of 2012, from which we compute the average demand for individual weekdays.

Table (22) provides the comparison results (on profit and lost demand) between our approach and the two benchmark approaches. Our approach is able to gain 3.5% in profit on average while the lost demand is reduced by an average of 45% over the practice of no repositioning during the day. In comparison with online heuristic, our approach is able to reduce the lost demand by an average of 41%, while the profit is increased by 3% on average. In the peak hours, our approach reduces the lost demand by an average of 46% and 44% over the static repositioning and online heuristic approach respectively.

We consider the trip history data of four quarters of 2013 for *Capital Bikeshare* and for each quarter we have done the same set of experiments. Table (23) shows that for the first quarter of data, our dynamic

approach is able to outperform static repositioning during the peak time as well as during the day, with respect to both the profit gain and the reduction in lost demand. We reduce the lost demand by an average of 20%, a significant improvement over the static repositioning. As expected, for all of these instances, the percentage gain in profit in the peak hours is much higher because most of the lost demand occur in the peak hours. Although the online heuristic performs well in the peak hours, it fails to provide a good quality solution when we consider a long planning horizon (38 time step). In case of long planning horizon, our approach outperforms the online heuristic for all the weekdays both in terms of profit gain and lost demand reduction.

	Whole day (5AM-12AM)				Peak period (5AM-12PM)			
	Gain over static repositioning		Gain over online heuristic		Gain over static repositioning		Gain over online heuristic	
	Profit gain	**Lost demand reduction**	**Profit gain**	**Lost demand reduction**	**Profit gain**	**Lost demand reduction**	**Profit gain**	**Lost demand reduction**
Mon	1.92%	20.61%	0.51%	5.43%	5.01%	29.07%	-0.74%	-2.81%
Tue	2.14%	19.82%	-0.18%	-0.46%	4.58%	23.08%	-4.31%	-20.19%
Wed	3.11%	24.59%	-0.11%	2.97%	8.89%	34.75%	0.39%	5.38%
Thu	3.55%	28.62%	5.28%	22.92%	7.17%	31.03%	2.18%	12.16%
Fri	3.34%	28.31%	3.39%	15.26%	7.56%	31.69%	2.51%	9.42%
Sat	0.04%	12.06%	-1.74%	-19.86%	-0.34%	12.15%	-2.5%	-10.59%
Sun	0.18%	9.44%	0.07%	-0.39%	-1.46%	3.33%	0.61%	13.86%
Mean	2.04%	20.49%	1.03%	3.70%	4.49%	23.59%	-0.27%	1.03%

Table 23: Profit and lost demand comparison (data set: *Capital Bikeshare*, 1st quarter of 2013).

	Whole day (5AM-12AM)				Peak period (5AM-12PM)			
	Gain over static repositioning		Gain over online heuristic		Gain over static repositioning		Gain over online heuristic	
	Profit gain	**Lost demand reduction**	**Profit gain**	**Lost demand reduction**	**Profit gain**	**Lost demand reduction**	**Profit gain**	**Lost demand reduction**
Mon	2.51%	27.1%	1.83%	18.76%	5.13%	32.84%	1.3%	9.57%
Tue	3.02%	26.44%	1.48%	15.33%	6.79%	28.72%	1.89%	8.84%
Wed	2.67%	22.09%	1.88%	14.31%	5.19%	21.83%	0.93%	3.46%
Thu	3.98%	32.45%	2.98%	24.02%	9.4%	38.19%	5.72%	24.51%
Fri	2.62%	30.76%	2.24%	22.75%	4.25%	27.41%	0.02%	0%
Sat	1.09%	16.52%	-0.23%	-3.72%	1.95%	28.08%	-0.93%	-8.96%
Sun	1.88%	25.65%	1.11%	10.96%	3.72%	40.2%	3.1%	20.78%
Mean	2.54%	25.86%	1.61%	14.63%	5.20%	31.04%	1.72%	8.31%

Table 24: Profit and lost demand comparison (data set: *Capital Bikeshare*, 2nd quarter of 2013).

Table (24) shows the percentage gain in profit and the percentage reduction in lost demand in comparison with the two benchmarks for the second quarter. Our approach is able to reduce the lost demand in all the cases by at least 16%, while the profit is improved by an average of 2.5% over the static repositioning. Our approach always almost outperforms the online heuristic also. We are able to reduce the lost demand by an average of 10%, while the profit is improved by an average of 1.5% in comparison with the online heuristic.

Table (25) shows the comparison with the two benchmarks for the third quarter. It is the busiest quarter in the year. For this quarter, our approach is able to reduce the lost demand by an average of 20%, while the profit is improved by an average of 3% over the static repositioning. Our approach also always performs better than the online heuristic for this quarter. Our approach is able to reduce the lost demand by an average of 13%, while the profit is improved by an average of 2% over the online heuristic. Moreover, these results show the strength of our approach in the presence of high customer demand.

	Whole day (5AM-12AM)				Peak period (5AM-12PM)			
	Gain over static repositioning		Gain over online heuristic		Gain over static repositioning		Gain over online heuristic	
	Profit gain	**Lost demand reduction**	**Profit gain**	**Lost demand reduction**	**Profit gain**	**Lost demand reduction**	**Profit gain**	**Lost demand reduction**
Mon	2.22%	20.81%	1.17%	9.62%	5.72%	27.11%	2.66%	10.2%
Tue	3.29%	25.79%	2.49%	18.01%	7.13%	29.45%	0.91%	5.79%
Wed	3.62%	28.06%	2.86%	21.29%	8.37%	34.59%	4.6%	21.43%
Thu	3.09%	30.58%	2.46%	23.05%	7.31%	35.89%	4%	21.1%
Fri	1.98%	26.69%	1.18%	13.45%	3.85%	26.43%	1.02%	2.97%
Sat	2.52%	31.18%	1.87%	17.25%	4.5%	49%	2.27%	21.12%
Sun	1.58%	26.95%	0.7%	8.54%	4.07%	40.38%	1.82%	14.59%
Mean	2.61%	27.15%	1.82%	15.89%	5.85%	34.69%	2.47%	13.89%

Table 25: Profit and lost demand comparison (data set: *Capital Bikeshare*, 3rd quarter of 2013).

	Whole day (5AM-12AM)				Peak period (5AM-12PM)			
	Gain over static repositioning		Gain over online heuristic		Gain over static repositioning		Gain over online heuristic	
	Profit gain	**Lost demand reduction**	**Profit gain**	**Lost demand reduction**	**Profit gain**	**Lost demand reduction**	**Profit gain**	**Lost demand reduction**
Mon	1.58%	15.82%	1.46%	12.86%	3.11%	19.32%	1.52%	9.09%
Tue	0.75%	12.96%	1.09%	3.75%	2.73%	19.4%	0.3%	2.14%
Wed	3.3%	25.53%	3.53%	23.38%	6.82%	29.37%	3.96%	17.12%
Thu	1.67%	16.48%	0.58%	3.99%	4.36%	22.49%	-1.27%	-5.97%
Fri	0.88%	15.98%	0.73%	1.85%	2.18%	19.28%	-1.92%	-6.08%
Sat	1.19%	12.48%	0.31%	-8.07%	0.26%	6.71%	-0.93%	-18.8%
Sun	0.29%	11.51%	-0.4%	-0.62%	-1.62%	-3.15%	-2.12%	-16.96%
Mean	1.38%	15.82%	1.04%	5.31%	2.55%	16.20%	-0.07%	-2.78%

Table 26: Profit and lost demand comparison (data set: *Capital Bikeshare*, 4th quarter of 2013).

Table (26) shows the percentage gain in profit and the percentage reduction in lost demand in comparison with the two benchmarks for the last quarter. For this data set, our approach reduces the lost demand by at least 12% over the static repositioning, while in comparison with the online heuristic our approach reduces the lost demand by an average of 5%. For all of these quarters, the percentage gain in profit in the peak hours is almost double because most of the lost demand occur during this period.

References

Barth, M., Todd, M., & Xue, L. (2004). User-based vehicle relocation techniques for multiple-station shared-use vehicle systems. In *Transportation Research Board Annual Conference CD-ROM*. Citeseer.

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

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

Bertsekas, D. P. (1999). *Nonlinear programming* (2nd edition). Athena Scientific.

Borgnat, P., Abry, P., Flandrin, P., Robardet, C., Rouquier, J.-B., & Fleury, E. (2011). Shared bicycles in a city: A signal processing and data analysis perspective. *Advances in Complex Systems*, *14*(03), 415–438.

Borgnat, P., Abry, P., Flandrin, P., & Rouquier, J.-B. (2009). Studying Lyon's Vélo'V: A statistical cyclic model. In *Proceedings of the European Conference on Complex Systems (ECCS)*. Complex System Society.

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

Coelho, L. C., Cordeau, J.-F., & Laporte, G. (2013). Thirty years of inventory-routing. *Transportation Science*, *48*(1), 1–19.

Contardo, C., Morency, C., & Rousseau, L.-M. (2012). Balancing a dynamic public bike-sharing system. Tech. rep., CIRRELT.

Dean, T., & Lin, S.-H. (1995). Decomposition techniques for planning in stochastic domains. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, Vol. 2, pp. 1121–1127.

Di Gaspero, L., Rendl, A., & Urli, T. (2013). A hybrid ACO+CP for balancing bicycle sharing systems. In *Hybrid Metaheuristics*, pp. 198–212. Springer, Berlin, Heidelberg.

Di Gaspero, L., Rendl, A., & Urli, T. (2016). Balancing bike sharing systems with constraint programming. *Constraints*, *21*(2), 318–348.

Erdoğan, G., Laporte, G., & Wolfler Calvo, R. (2014). The static bicycle relocation problem with demand intervals. *European Journal of Operational Research*, *238*(2), 451–457.

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

Fishman, E., Washington, S., & Haworth, N. L. (2014). Bike share's impact on car use: Evidence from the United States, Great Britain, and Australia. *Transportation Research Part D: Transport and Environment*, *31*, 13–20.

Fricker, C., & Gast, N. (2016). Incentives and redistribution in homogeneous bike-sharing systems with stations of finite capacity. *EURO Journal on Transportation and Logistics*, *5*(3), 261–291.

Froehlich, J., Neumann, J., & Oliver, N. (2008). Measuring the pulse of the city through shared bicycle programs. In *Proceedings of the International Workshop on Urban, Community, and Social Applications of Networked Sensing Systems (UrbanSense)*, pp. 16–20.

Furmston, T., & Barber, D. (2011). Lagrange dual decomposition for finite horizon Markov decision processes. In *Proceedings of the Machine Learning and Knowledge Discovery in Databases: European Conference (ECML PKDD)*, pp. 487–502.

George, D. K., & Xia, C. H. (2011). Fleet-sizing and service availability for a vehicle rental system via closed queueing networks. *European Journal of Operational Research*, *211*(1), 198–207.

Geramifard, A., Doshi, F., Redding, J., Roy, N., & How, J. (2011). Online discovery of feature dependencies. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 881–888.

Ghosh, S., & Varakantham, P. (2016). Strategic planning for setting up base stations in emergency medical systems. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*, pp. 385–393.

Giunchiglia, F., & Walsh, T. (1992). A theory of abstraction. *Artificial Intelligence*, *57*(2), 323–389.

Gordon, G. J., Varakantham, P., Yeoh, W., Lau, H. C., Aravamudhan, A. S., & Cheng, S.-F. (2012). Lagrangian relaxation for large-scale multi-agent planning. In *Proceedings of the The IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, Vol. 2, pp. 494–501.

Guestrin, C., & Gordon, G. (2002). Distributed planning in hierarchical factored MDPs. In *Proceedings of the Uncertainty in artificial intelligence (UAI)*, pp. 197–206.

Hernández-Pérez, H., & Salazar-González, J.-J. (2004). Heuristics for the one-commodity pickup-and-delivery traveling salesman problem. *Transportation Science*, *38*(2), 245–255.

Kabra, A., Belavina, E., & Girotra, K. (2015). Bike-share systems: accessibility and availability. Tech. rep., Chicago Booth No. 15-04.

Kek, A. G., Cheu, R. L., Meng, Q., & Fung, C. H. (2009). A decision support system for vehicle relocation operations in carsharing systems. *Transportation Research Part E: Logistics and Transportation Review*, *45*(1), 149–158.

Knoblock, C. (1993). *Generating abstraction hierarchies: An automated approach to reducing search in planning*, Vol. 214. Springer Science & Business Media, New York.

Knoblock, C. A. (1991). Search reduction in hierarchical problem solving. In *AAAI Conference on Artificial Intelligence (AAAI)*, pp. 686–691.

Kök, A. G., & Fisher, M. L. (2007). Demand estimation and assortment optimization under substitution: methodology and application. *Operations Research*, *55*(6), 1001–1021.

Kumar, A., Singh, S., Gupta, P., & Parija, G. (2014). Near-optimal nonmyopic contact center planning using dual decomposition. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*, pp. 395–403.

Kumar, A., Wu, X., & Zilberstein, S. (2012). Lagrangian relaxation techniques for scalable spatial conservation planning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pp. 309–315.

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

Laporte, G., Meunier, F., & Wolfler Calvo, R. (2015). Shared mobility systems. *4OR*, *13*(4), 341–360.

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

Lees-Miller, J. D., Hammersley, J. C., & 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. In *https://hal.archivesouvertes.fr/hal-00757228*.

Li, L., Walsh, T. J., & Littman, M. L. (2006). Towards a unified theory of state abstraction for MDPs. In *Proceedings of the International Symposium on Artificial Intelligence and Mathematics (ISAIM)*, pp. 1–10.

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

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

Lu, T., & Boutilier, C. (2015). Value-directed compression of large-scale assignment problems. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pp. 1182–1190.

Mahadevan, S. (2002). Spatio-temporal abstraction of stochastic sequential processes. In *Proceedings of the Abstraction, Reformulation, and Approximation: International Symposium (SARA)*, pp. 33–50.

Martinez, L. M., Caetano, L., Eiró, T., & Cruz, F. (2012). 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*, 513–524.

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

Musalem, A., Olivares, M., Bradlow, E. T., Terwiesch, C., & Corsten, D. (2010). Structural estimation of the effect of out-of-stocks. *Management Science*, *56*(7), 1180–1197.

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

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

O'Mahony, E., & Shmoys, D. B. (2015). Data analysis and optimization for (Citi) bike sharing. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pp. 687–694.

Papageorgiou, D. J., Nemhauser, G. L., Sokol, J., Cheon, M.-S., & Keha, A. B. (2014). MIRPLib - A library of maritime inventory routing problem instances: Survey, core model, and benchmark results. *European Journal of Operational Research*, *235*(2), 350–366.

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

Raidl, G. R., Hu, B., Rainer-Harbach, M., & Papazek, P. (2013). Balancing bicycle sharing systems: Improving a VNS by efficiently determining optimal loading operations. In *Proceedings of th Hybrid Metaheuristics: International Workshop (HM)*, pp. 130–143.

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

Raviv, T., Tzur, M., & 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.

Rogers, D. F., Plante, R. D., Wong, R. T., & Evans, J. R. (1991). Aggregation and disaggregation techniques and methodology in optimization. *Operations Research*, *39*(4), 553–582.

Saisubramanian, S., Varakantham, P., & Chuin, L. H. (2015). Risk based optimization for improving emergency medical systems. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pp. 702–708.

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

Shu, J., Chou, M., Liu, Q., Teo, C.-P., & Wang, I.-L. (2010). Bicycle-sharing system: deployment, utilization and the value of re-distribution. In *Technical Report, NUS. http://bschool.nus.edu/Staff/bizteocp/BS2010.pdf*.

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

Singla, A., Santoni, M., Bartók, G., Mukerji, P., Meenen, M., & Krause, A. (2015). Incentivizing users for balancing bike sharing systems. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pp. 723–729.

Sturtevant, N. R., & White, A. M. (2006). Feature construction for reinforcement learning in hearts. In *Proceedings of the International Conference on Computers and Games (CG)*, pp. 122–134.

Sutton, R. S., Precup, D., & Singh, S. (1999). Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence*, *112*(1), 181–211.

Vulcano, G., Van Ryzin, G., & Ratliff, R. (2012). Estimating primary demand for substitutable products from sales transaction data. *Operations Research*, *60*(2), 313–334.

Yue, Y., Marla, L., & Krishnan, R. (2012). An efficient simulation-based approach to ambulance fleet allocation and dynamic redeployment. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pp. 398–405.